

#86 December 1983

\$2.95 [\$3.50 in Canada]

Dr. Dobb's Journal

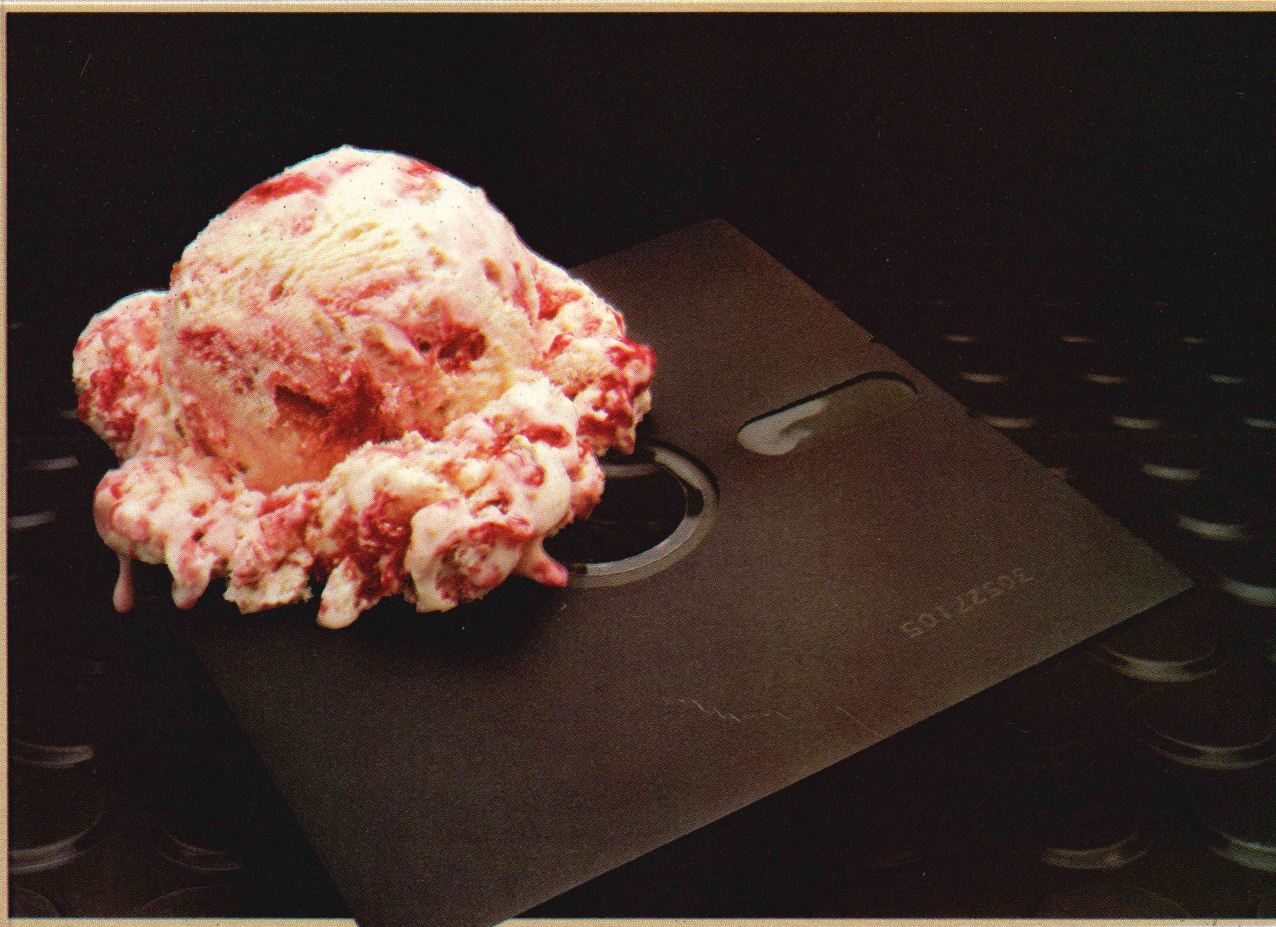
For Users of Small Computer Systems



**Faster Circles
For Apples**

**Dysan's
Digital
Diagnostic
Diskette**

**Cursor Control
For Dumb
Terminals**



NOW THERE ARE TWO WAYS TO MAKE dBASE II A TREAT. (ONLY ONE IS GUARANTEED.)

Let Autocode write your programs for you.

For those of you with no desire to join the ranks of computer programmers just to get your work done, treat yourself to Autocode. Autocode is the easiest, fastest way to get what you want from dBase II, because it generates complete, high quality programs from simple instructions. Whether you're a dBase II master or just learning, you'll like the way Autocode writes powerful, elegant code with a minimum of effort. And you'll appreciate the fact that Autocode, unlike slower, less capable program generators, supports sophisticated dBase II features like on-screen calculations and multi-tiered criteria for report generation.

dBase II is a registered trademark of Ashton-Tate.
Autocode I is a trademark of Stemmos Ltd.

Even if you're an experienced programmer, the consistent quality of Autocode's programs will win you over.

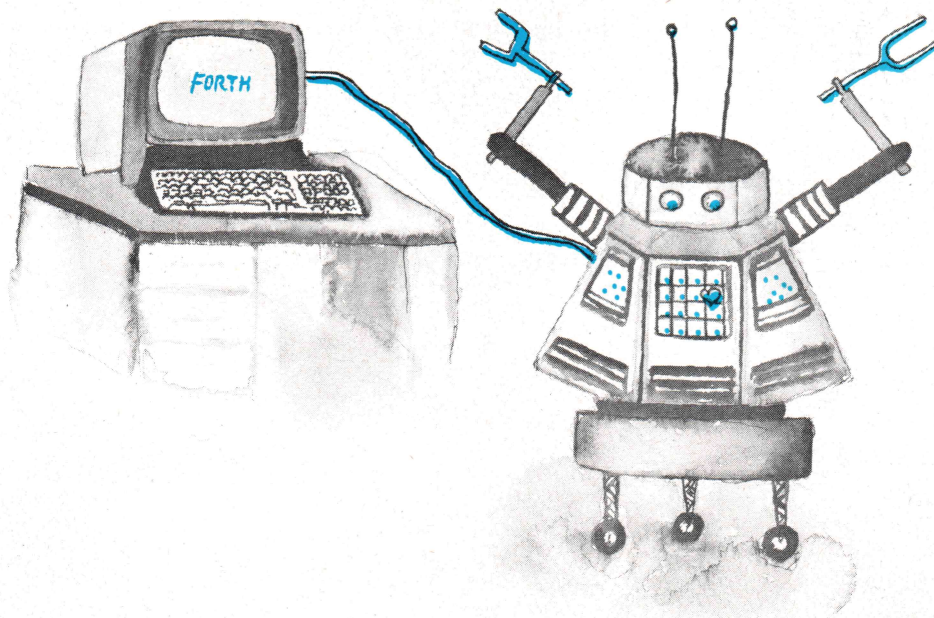
Use Autocode without risk.

Give it a try. For only \$195.00, you get the unabridged Autocode disk and our concise, entertaining, clearly written manual. We're so sure you'll like the simplicity of Autocode's easy to use menu-driven operation, the only risk is that you'll never go back to dBase II alone. So for a taste of the best thing that's ever happened to dBase II, try Autocode. Contact your dealer, or call toll-free 800-262-8800 or (415) 777-3800. Or write Axel Johnson Corporation, 666 Howard St., San Francisco, California 94105.

Autocode I

☆ A member of the Axel Johnson Group.

FORTH gives you TOTAL CONTROL.



GRAPHICS • GAMES • COMMUNICATIONS ROBOTICS • DATA ACQUISITION • PROCESS CONTROL

FORTH: FOR Z-80®, 8086, 68000, and IBM® PC

- **FORTH** programs are instantly portable across the four most popular microprocessors.

- **FORTH** is interactive and conversational, but 20 times faster than BASIC.

- **FORTH** programs are highly structured, modular, easy to maintain.

- **FORTH** affords direct control over all interrupts, memory locations, and i/o ports.

- **FORTH** allows full access to DOS files and functions.

- **FORTH** application programs can be compiled into turnkey COM files and distributed with no license fee.

- **FORTH** Cross Compilers are available for ROM'ed or disk based applications on most microprocessors.

FORTH Application Development Systems include interpreter/compiler with virtual memory management and multi-tasking, assembler, full screen editor, de-compiler, utilities, and 130 + page manual. Standard random access files used for screen storage, extensions provided for access to all operating system functions.

Z-80 FORTH for CP/M® 2.2 or MP/M II, \$50.00; **8080 FORTH** for CP/M 2.2 or MP/M II, \$50.00; **8086 FORTH** for CP/M-86 or MS-DOS, \$100.00; **PC/FORTH™** for PC-DOS, CP/M-86, or CCPM, \$100.00; **68000 FORTH** for CP/M-68K, \$250.00.

FORTH + Systems are 32 bit implementations that allow creation of programs as large as 1 megabyte. The entire memory address space of the 68000 or 8086/88 is supported directly for programs and data.

PC FORTH + \$250.00

8086 FORTH + for CP/M-86 \$250.00

68000 FORTH + for CP/M-68K \$400.00

Nautilus Cross Compiler allows you to expand or modify the FORTH nucleus, recompile on a host computer for a different target computer, generate headerless and ROM-able code. Supports forward referencing. Produces executable image in RAM or disk file. No license fee for applications. Pre-requisite: Application Development System for host computer, \$300.00.

FORTH Native Code Compiler, requires Z-80 FORTH, CP/M 2.2, \$100.00.

Extension Packages

Software floating point (Z-80, 8086, PC only), \$100.00; AMD 9511 support (Z-80, 8086, 68000 only), \$100.00; Intel 8087 support (8086, PC only), \$100.00; Advanced color graphics (PC only), \$100.00; Symbolic interactive debugger (PC only), \$100.00; PC/TERM Communications/file transfer for Smartmodem, \$60.00; Cross reference utility, \$25.00; PC/GEN (custom character sets, PC only), \$50.00; Curry FORTH Programming Aids, \$150.00; Hierarchical file manager, \$50.00; B-Tree index manager, \$125.00; B-Tree index and file manager, \$200.00; QTF + Screen editor for IBM PC, \$100.00;

AUGUSTA, Ada subset compiler from Computer Linguistics for Z-80 CP/M 2.2 systems, \$90.00.

"Starting FORTH" tutorial by Brodie, soft-cover, \$16.00.

INTEL 8087-3 Numeric Coprocessor, \$250.00.

83 - Standard version of all application development systems available soon. All registered users will be entitled to software update at nominal cost.

Z-80 and 8080 **FORTH** require 48 Kbytes RAM. 8086 and 68000 **FORTH** require 64 Kbytes RAM. Disk formats available include: 8" standard CP/M SSSD, Northstar 5 1/4" QD, Kaypro 5 1/4", Apple 5 1/4", Micro-Mate 5 1/4", MS-DOS 5 1/4", Osborne 5 1/4" DD, and Victor 9000. Most other formats can be special ordered.

Z-80 is a registered trademark of Zilog, Inc.; CP/M is a registered trademark of Digital Research, Inc.; IBM is a registered trademark of International Business Machines Corp.; Augusta is a trademark of Computer Linguistics; dBASE II is a trademark of Ashton-Tate; PC/FORTH and PC/GEN are trademarks of Laboratory Microsystems Inc.



Laboratory Microsystems Incorporated
4147 Beethoven Street, Los Angeles, CA 90066
Phone credit card orders to (213) 306-7412

Circle no. 55 on reader service card.

See us at

SOFTCON
La Superdome
New Orleans
Tuesday-Thursday
February 21-23, 1984

FAMOUS GIANT KILLERS THROUGHOUT HISTORY

David, Jack & PATTM

Size has always been a most formidable foe. Tales of David and Goliath and Jack and the Beanstalk are clear examples of how "Bigger is not always better". In fact, throughout history, all great giant killers have had one thing in common... small size but a surprisingly big capability.

And now, there's PAT, Dysan's unique Performance and Alignment tester that cuts giant floppy drive problems down to size.

PAT is the first hand-held intelligent tester for flexible diskette drives. Used in conjunction with Dysan's Digital Diagnostic Diskette (DDD)TM, PAT's versatile and precise testing capabilities can test drive alignment and performance criteria. Plus, PAT also checks read/write data handling functions and serves as an

exerciser without the use of additional equipment, all with minimal operator instruction.

And best of all, PAT sells for a small fraction of what the big testers cost.

PAT's Giant-Size Capabilities Include:

- Head radial alignment
- Head positioner linearity
- Head azimuth alignment
 - Index timing
- Head positioner skew
 - Spindle speed
- Read/write verification
- Seek test verification

For quality assurance, field service, incoming inspection, engineering evaluation and field disk drive sales, discover PAT. Another big difference from Dysan.



dysan **Dysan**[®]
CORPORATION

Corporate Headquarters
5201 Patrick Henry Drive
Santa Clara, CA 95050
(800) 551-9000
(408) 988-3472

For more information on PAT,
call toll free (800) 551-9000 or (408) 988-3472.
Ask about our limited-time bonus offer!

Circle no. 35 on reader service card.

Publisher — Jane Nissen Laydley
Editor — Reynold Wiggins
Managing Editor — Craig LaGrow
Contributing Editors —
Robert Blum, Dave Cortesi,
Ray Duncan, Anthony Skjellum,
Michael Wiesenber
Marketing Director — Beatrice Blatteis
Marketing Assistant — Sally Brenton
Marketing Coordinator — Ed Gueble
Advertising Director — Beatrice Blatteis
Advertising Sales —
Alice Hinton, Walter Andrzejewski
Circulation Director — Terri Pond
Circulation Assistant — Billie Greenwood
Production Manager — Barbara Ruzgerian
Production Assistants —
Shelley Rae Doeden, Alida Hinton,
Tom Martin
Typesetter — Paula Fairchild
Cover — Alida Hinton

Copyright © 1983 by People's Computer Company unless otherwise noted on specific articles. All rights reserved.

Subscription Rates: \$25 per year within the United States; \$44 for first class to Canada and Mexico; \$62 for airmail to other countries. Payment must be in U.S. Dollars, drawn on a U.S. Bank.

Writer's Guidelines: All items should be typed, double-spaced on white paper. Listings should be produced by the computer, using a *fresh, dark ribbon* on continuous white paper. Please avoid printing on perforations. Payment is in contributor's copies. Requests to review galley must accompany the manuscript when it is first submitted. Authors may receive a copy of the complete writer's guidelines by sending a self-addressed, stamped envelope.

Donating Subscribers Contributing Subscriber: \$50/year (\$25 tax deductible). Retaining Subscriber: \$75/year (\$50 tax deductible). Sustaining Subscriber: \$100/year (\$75 tax deductible). Lifetime Subscriber: \$1000 (\$800 tax deductible). Corporate Subscriber: \$500/year (\$400 tax deductible, receives five one-year subscriptions).

Contributing Subscribers: Christine Bell, W. D. Rausch, DeWitt S. Brown, Burks A. Smith, Robert C. Luckey, Transdata Corp., Mark Ketter, John W. Campbell, Friden Mailing Equipment, Frank Lawyer, Rodney Black, Thomas Davis, Kenneth Drexler, Real Paquin, Ed Malin, John Saylor, Jr., Ted A. Reuss III, Infoworld, Stan Veit, Western Material Control, S. P. Kennedy, John Hatch, Richard Jorgensen, John Boak. **Lifetime Subscriber:** Michael S. Zick.

Foreign Distributors **UK & Europe:** Homecomputer Vertriebs HMBH 282, Flugelstr. 47, 4000 Dusseldorf 1, West Germany; La Nacelle Bookstore, Procedure D'Abonnement 1-74, 2, Rue Campagne — Premiere, F-75014 Paris, France; Computercollectief, Amstel 312A, 1017 AP Amsterdam, Netherlands. **Asia & Australia:** ASCII Publishing, Inc., 4F Segawa Bldg. 5-2-2, Jingumae, Shibuya-Ku, Tokyo 150, Japan; Computer Services, P.O. Box 13, Clayfield QLD 4011, Australia; Computer Store, P.O. Box 31-261, 22B Milford Rd., Milford, Auckland 9, New Zealand. (Write for Canadian distributors)

Dr. Dobb's Journal

For Users of Small Computer Systems

December 1983
Volume 8, Issue 12

CONTENTS

ARTICLES

18 Faster Circles for Apples

by Myron L. Pulier

Inspired by Daniel Lee's "Fast Circle Routine" (May 1983 *DDJ*, No. 79), these circle-generation programs use Applesoft BASIC and assembly language to provide greater speed for Apple users. Two other *DDJ* readers also submit their amplifications on Lee's algorithm.

32 Cursor Control for Dumb Terminals

by Ian Ashdown

Want to teach your "antique" terminal to do direct cursor addressing? This author has developed a conversion program that's easy to implement on your CP/M system.

40 Dyan's Digital Diagnostic Diskette

by Loren Amelang

Tired of paying someone else a bundle of money to analyze and repair your faulty disk drive? Let this author show you how a program and Dyan's DDD can help you make your own diagnoses, without an oscilloscope or other fancy equipment.

80 Building a Programmable Frequency Synthesizer

by Michael L. Simon

Here's how one author used a phase-locked loop to create a programmable frequency synthesizer. Using a spare counter/timer channel on a single-board computer, the configuration can multiply frequencies exactly.

DEPARTMENTS

9 Editorial

9 Letters

12 Dr. Dobb's Clinic

A brief look at two new computer columnists

14 C/Unix Programmer's Notebook

How Unix-type environments can lead to *non*-interactive and user-unfriendly software

92 CP/M Exchange

Solutions to interfacing a hard disk within a CP/M environment, and how to speed up SUBMIT file truncation

102 16-Bit Software Toolbox

A look at the new MS-DOS EXEC function — a powerful "Unix-like" modification which should work well in a multi-tasking environment

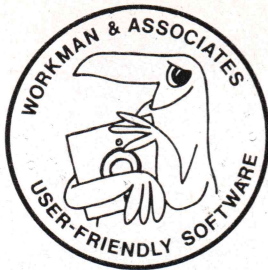
112 Software Reviews

116 Book Reviews

120 Of Interest

126 Advertiser Index

Dr. Dobb's Journal (USPS 307690) is published twelve times per year by People's Computer Company, P.O. Box E, Menlo Park, CA 94026. Second class postage paid at Menlo Park, California 94026 and additional entry points. Address correction requested. Postmaster: send form 3579 to People's Computer Company, Box E, Menlo Park, CA 94026 • 415/323-3111 **ISSN 0278-6508**



WORKMAN & ASSOCIATES

112 Marion Avenue
Pasadena, CA 91106
(213) 796-4401

WRITE

The Writer's Really Incredible Text Editor lives up to its name! It's designed for creative and report writing, and carefully protects your text. Includes many features missing from WordStar, such as sorted directory listings, fast scrolling, and trial printing to the screen. All editing commands are single-letter and easily changed. WRITE is \$239.00.

"A PRIMER ON PASCAL FOR CP/M SYSTEMS"

If you want to learn Pascal/M or MT+ on your own, this is the way. The Primer contains sample programs, suggestions, and notes on disk and in the manual. Programs come from the books "Programming in Pascal" and "Software Tools in Pascal". Discounted to only \$64.50!

All US orders are postpaid. We ship from stock on many formats, including: 8", Apple, NorthStar, Osborne, KayPro, Monroe, Otrona, Epson QX-10, NEC/PC, DEC VT-180, TI Professional, Access, Morrow. Please request our new catalog. We welcome COD orders.

Circle no. 128 on reader service card.

AZTEC C — 'C' PROGRAM DEVELOPMENT SYSTEM PORTABLE SOFTWARE APPLE CP/M IBM

WHY PROFESSIONALS CHOOSE AZTEC C

MANX SOFTWARE SYSTEMS

- AZTEC C is the most complete implementation of UNIX V7 'C' available for microcomputers.
- AZTEC C is portable. 'C' code can be freely moved between UNIX V7 and microcomputer systems. AZTEC C is available for PC DOS (MSDOS), CP/M 86 (MP/M 86), CP/M 80 (MP/M 80), APPLE DOS, MODEL III (IV), COMMODORE 64, and ATARI. All versions of AZTEC C are source compatible.
- AZTEC C is a complete development system that includes relocating assembler, linkage editor, library utility, debugging aids, overlay support, interfaces to Microsoft and Digital Research development software, and run time routines for I/O, utility, and scientific functions. Source for all library routines is provided.
- AZTEC C generates fast native code for the 6502, 8080, Z80, 8088, and 8086.
- Cross compilers are available from PDP-11, 8086, 8080, and 6502 processors.
- Since its release in 1981, AZTEC C has been acquired by several thousand users. Commercial applications include a wide variety of business, scientific, word processing, database, entertainment and financial applications.

PRICE LIST

AZTEC C86	PC DOS (MSDOS)	\$249
AZTEC C86	CP/M 86 (MP/M 86)	249
AZTEC C86	CP/M 80 (MP/M 80)	199
AZTEC C86	APPLE DOS	199
AZTEC C65	COM 64 OR ATARI	199
AZTEC C65	COM 64 OR ATARI	2,000
AZTEC Cross Compilers	PDP-11	500
AZTEC Cross Compilers	Other	500

Order by phone or mail. Specify product and disk format. Check, Money Orders, COD, VISA, MC, and purchase orders are acceptable. NJ add 6% sales tax.

MANX SOFTWARE SYSTEMS
Box 55, Shrewsbury, NJ 07701
(201) 780-4004
(201) 530-7997
(201) 530-7708

Order phone:
Tech information:
Tech support:

Shipping: COD, 2nd day delivery, or Canada, add \$5. Canada 2nd day or US next day delivery, add \$20. Outside North America, add \$20, and for 2nd day add \$75.

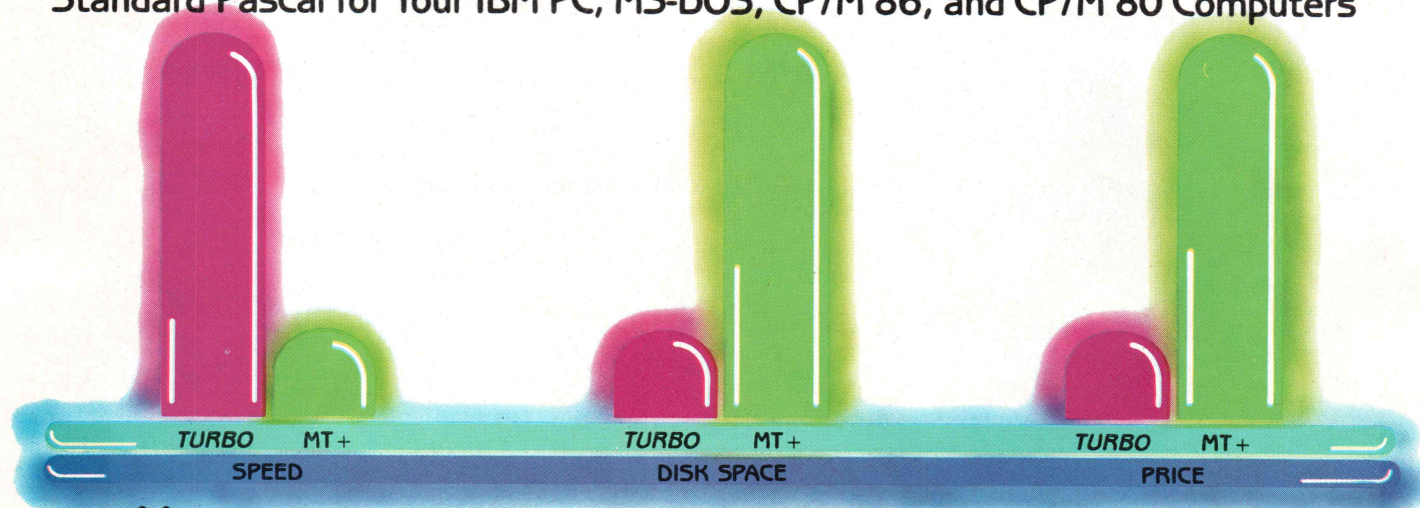
Circle no. 61 on reader service card.

PASCAL

\$49.95

Introductory
Offer

Standard Pascal for Your IBM PC, MS-DOS, CP/M 86, and CP/M 80 Computers



Hang on to your seats! It's Turbo Pascal.

	Turbo Pascal	JRT	MT+
8 & 16 bit	YES	NO	YES
Editor	YES	NO	NO
Generate Object Code	YES	NO	YES
Locates RunTime Errors Directly in Source Code	YES	NO	NO
Compilation Speed *	1 s.	46 s.	69 s. includes linking
Execution Speed *	6 s.	69 s.	8 s.
Disk Space	28K including editor	85K + editor	168K + editor
Price	\$49.95	\$39.95	\$595.00

There has never been a Pascal compiler this good with so many powerful features. We know what you've been waiting for: a true Pascal compiler that works fast, offers a full screen editor, and has a great price.

Turbo Pascal has it all. First, we've included a built-in, interactive full screen, Wordstar compatible editor; it not only lets you correct errors, but during program compilation the cursor even jumps *directly* to the error and waits for your correction. No kidding. Second, it takes only 28K of disk space, including the editor; and on your microcomputer you need all the space you can get. **Turbo Pascal** is

10 to 70 times faster during compilation, as well as execution than Digital Research's MT+ or JRT Pascal.

Hard to believe your good fortune on the price? Don't worry. We're Borland, and we produce only quality, state-of-the-art software. Companies such as Micro Pro, Morrow Computers, Access and others distribute our software products, so you can't go wrong.

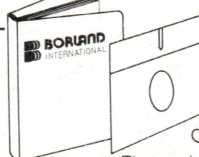
Place your order today. And we'll ship your **Turbo Pascal** out fast. For VISA and MasterCard orders call toll free:

1-800-227-2400 X 968
IN CA: 1-800-772-2666 X 968

Turbo Pascal \$49.95 + \$2.00 shipping per copy.

Check ☐ Money Order ☐
VISA ☐ MasterCard ☐

Card #: _____
Exp date: _____ Shipped UPS.



My system is: 8 bit ☐ 16 bit ☐
Operating system: CP/M 80 ☐
CP/M 86 ☐ MS DOS ☐ PC DOS ☐
Computer: _____ Disk Format: _____

Please be sure model number and format are correct.

NAME: _____
ADDRESS: _____
CITY/STATE/ZIP: _____
TELEPHONE: _____

BORLAND INTERNATIONAL

Borland International
4807 Scotts Valley Drive
Scotts Valley, California 95066

Dealer and Distributor inquiries welcome.

California residents add 6½% sales tax. Outside North America add \$15.00 for airmail, or \$5.00 for surface mail. Checks must be on a U.S. bank, and in U.S. dollars. Sorry, no C.O.D.

Benchmark data based on EightQueens in "Algorithms + Data Structures = Programs" by N. Wirth (Prentice-Hall, publisher)

Turbo Pascal is a trademark of Borland International. MT+ is a trademark of MT MicroSystems. JRT Pascal is a product of JRT. Wordstar is a trademark of Micropro.

Q: What's easier than spending money?

A: Keeping track of it with... **Checks & Balances**

At last! A *full screen* editing checking and finance program so informative and easy to use you'll wonder how you lived without it! *Especially at \$49⁹⁵*

Sixteen user-definable categories.

Category totals instantly updated with each new entry or change.

Screen always shows four entries. Use your cursor to add or change any entry anytime. Scroll forward and back through your register or instantly go to any entry.

Includes categories for bills, cash expenditures, deposits and service charges.

Checks and Balances aligns decimals and left justifies text.

For complete records, cash expenditures are included in category totals but not bank balance.

Entries which have cleared through your bank.

All commands in plain English.

True checking balance instantly updates with each entry.

Bank statement balance reflects only items checked off as having cleared your bank.

```

4389.90 A =ADVERTISING      83.48 O =OFFICE SUPPLIES
253.67 LABR =CONTRACT LABOR  36.37 P =PHONE
4139.23 DEP =DEPOSITS       160.70 POST=POSTAGE AND SHIPPING
132.51 ENTR =ENTERTAINMENT   806.80 REF =REFUNDS
2954.86 ER =EQUIPMENT FOR RESALE  527.89 R =ROYALTIES
3440.12 E =EQUIPMENT-IN HOUSE  221.27 TAX =TAXES-SRA AND OTHERS
1370.88 DEPM =MERCHANT DEPOSITS    0.00 T =TRAVEL EXPENSES
480.60 MISC =MISC. EXPENSES    45.03 U =UTILITIES
0.00 BILL      132.51 *CASH    5510.11 DEP      0.00 SPCH
=====
# 1317 06/14/83 Pay TO: MARVIN JONES      $    55.00 X
  (A)  Memo: PRINTING & LABOR
# 1318 06/10/83 Pay TO: ALLENBACH INDUSTRIES $    267.20 X
  (ER) Memo: INVOICE #7830, 7904
#CASH 06/10/83 Pay TO: TEMPURA INN      $    132.51
  (ENTR) Memo: MEETING TO DISCUSS NEW ADVERTISING
#DEP  05/21/83 Pay TO: DEPOSIT           $    240.00 X
  (DEP) Memo: SALES FROM MAGAZINE ADD
1.....1.....2.....3.....4.....5.....6.....7.....8
File=CDE = DISPLAY & EDIT = TRUE $ 12209.42 BANK $ 7750.41
SHOW
  
```

So easy to use! See your financial totals start to add up before your eyes in less than five minutes! Checks and Balances' cursor can move anywhere, anytime, to add or change entries, and all totals will instantly be updated on your screen when you change an amount or move a check to a different category. Checks and Balances gives you complete printed reports of the entire register, each category, or reports based on any entry in "Pay to" or "Memo"—even bills and uncashed or missing checks. File size is limited only by your disk capacity. Over 32,000 entries are possible.

Technical: 80 x 24 CRT with addressable cursor, reverse video optional. Z-80 processor, one or more disk drives with over 180k capacity. 56k RAM.

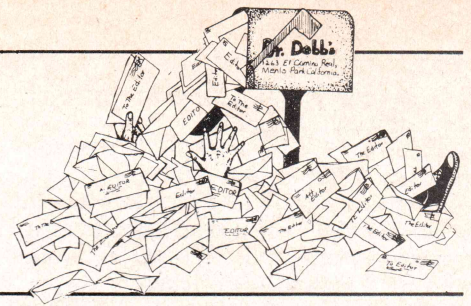


CDE SOFTWARE
2463 McCready Avenue
Los Angeles, CA 90039



(213) 661-2031

Circle no. 14 on reader service card.



Augusta Q & A

Dear DDJ:

Edward Mitchell's four-part series on his Augusta compiler, concluded in the July 1983 issue, was certainly interesting. The sheer magnitude of effort that he put into it is mind-boggling. It was definitely a plus that he gave a detailed description of his program, rather than just a long listing.

But I find myself baffled by his penchant for doing things the hard way. He rightly points out that BASIC is a less than ideal language for compiler writing. OK, he couldn't find a version of Pascal, C, or Fort to run on his Osborne. (Really??) But since he's obviously interested in systems programming, why on earth use an Osborne?

Along the same lines, one wonders why he chose to use a recursive descent parser. When coded in a language that permits recursive subroutine calls, one can write such a parser simply, rapidly, and bug-free. But BASIC permits no such thing. Since he had to go to the trouble of maintaining a push-down stack to simulate recursion, why not write an augmented push-down machine parser? (See P. M. Lewis II, D. J. Rosenkrantz, and R. E. Stearns, *Compiler Design Theory*, Addison-Wesley, 1976.) Such an algorithm requires maintaining a stack of only the symbols in the same LL(1) grammar used for recursive descent parsing. Logically, the two algorithms are equivalent since each subroutine call in recursive descent corresponds to a symbol in the

grammar. By doing "recursive descent" in BASIC, he is saddled with stacking subroutine calls and local variables — making his stack larger than necessary when memory is at a premium. In addition, the additional work needed to simulate recursion complicated his code and probably added enormously to the debugging time. A table-driven augmented push-down machine parser would have been shorter and simpler.

While I'm impressed by his having actually gotten Augusta up and running, it seems to be a classic example of an unnecessarily complicated solution to a problem resulting from choosing the wrong design and working with the wrong tools. I wouldn't recommend to others to follow that lead.

Very truly yours,
Clyde B. Schechter
520 West 122 Street
New York, NY 10027

Reply to Clyde Schechter:

Clyde Schechter raises some questions that other readers may also be curious about so I'll try to respond as best I can.

To put the Augusta project in proper perspective: Augusta was begun as a hobbyist project. That means it was done essentially for the sake of doing it, in the sense that rock climbers attempt difficult climbs solely for the sake of the climb.

Obviously, as I explained in Part IV of the Augusta series, neither BASIC nor

an Osborne I was the ideal development vehicle for a large software project. But in view of the hobbyist nature of Augusta's development, purchasing a "real" computer was out of the question.

Languages other than BASIC were available for the Osborne I. But due to the 92K disk size on the Osborne I, the other languages were effectively useless. At least one of the languages required almost 140K for the compiler alone. That leaves little room for source code. And since the Augusta compiler had yet to be written, I could only estimate how much disk space I would need for source and object modules.

As I mulled over the choices, Microsoft BASIC seemed to be the best, a small program itself, with both interactive and compiled versions available. In addition, Tiny-Pascal, a project similar in scope to Augusta, had also been written in BASIC, so I knew that it was "do-able." In fact, the bulk of Augusta was written in about five weeks, working only part time. Once compiled with the BASIC compiler, Augusta was compiling at over 300 lines per minute, which compares quite favorably to other microcomputer compilers. So all in all, it really wasn't that difficult to write the compiler in BASIC. (The greatest difficulties were with the p-code interpreter which required several months of writing and tweaking.)

Mr. Schechter correctly observes that it's relatively easy to write a recursive descent parser in a language that permits

EDITORIAL

Here at DDJ, as we reach the end of 1983 and look back over the past twelve issues, we are encouraged by what we see. The number of pages is roughly twice what it was a year ago. Circulation has increased substantially. The new faces that have appeared in the columns have brought with them fresh enthusiasm and perspectives. The somewhat altered face of the magazine makes it a bit easier to find on the newsstands, and also a bit more readable. Our group of technical referees has been expanded, and authors are now being paid.

In the midst of all this change, we have fought to maintain our focus. Your responses to our recently completed reader survey have given us additional insight into what we are doing right and what we can improve, what you want and what you are doing. Our thanks to all those who responded.

Also helpful throughout the year has been the Editorial Response insert card. Please continue to use it to vote for your favorite item in the issue or to drop us short comments. The Doctor is never too busy to listen, and the postage is covered.

As we leave 1983, we look for 1984 to be one of the best years yet for DDJ. All of us here are rededicating ourselves to the task of improving the *Journal*. Our thanks to all of you for your continuing support. We wish you a happy holiday season, and look forward to serving you in the New Year.

Reynold Wiggins

recursive subroutine calls. Prior to writing Augusta I had worked on both a recursive descent parser for a BASIC interpreter (written in Pascal) and a table-driven LL parser for a subset of Pascal (written in Simula). Based on that experience, I found recursive descent to be easier to work with. In addition, table-driven predictive parsers are more difficult to implement in a one-pass compiler, and a one-pass compiler was an important design goal.

To sum up, his points are well taken. Undoubtedly it could have been done easier with a different set of tools. However, the overall development of the Augusta compiler was not as difficult as some may imagine it to be. Further, my plans all along have been to rewrite Augusta in itself. Having the original compiler as a model simplifies the rewrite, since the new compiler is also based on recursive descent parsing.

As of this writing, most of Augusta has been rewritten in Augusta. Unfortun-

nately, I have been busy with other projects, including writing *Software Building Blocks for the IBM PC* (Hayden, 1984), as coauthor of Software Publishing Corporation's PFS:WRITE, and the teaching of a college summer school class. I will, however, be back working on Augusta in the near future.

I do appreciate everyone's comments, critical or praiseworthy. Thanks for writing.

Edward Mitchell

S-100 to Winchester Improved

Dear Editor,

David Cortesi has suggested some changes to my Winchester interface board (*Dr. Dobb's Journal*, October 1983) which would extend the range of CPU clock frequencies with which it would work. As an added benefit, the use and adjustment of the one-shots are eliminated.

Note that the modified schematic (Figure 1, below) includes a wait state

generator, capable of generating one or two wait states, either of which may be selected by appropriately connecting the jumper. Detailed information about wait state generators may be found in Chapter 5 of *Interfacing to S/100-IEEE 696 Microcomputers* by Sol Libes and Mark Garetz, Osborne/McGraw-Hill, 1981. If additional wait states are needed, say for an 8 MHz microprocessor, then the D-type flip-flop could be replaced by a shift register. Details about the use of such a device are to be found in the same book.

Sincerely yours,

Oscar Goldman, Professor
Mathematics Department
University of Pennsylvania
Philadelphia, PA 19104

Recovering Lost WordStar Text

Dear DDJ,

Every now and then, for various reasons, I lose what I have just typed into WordStar. This induces a case of helpless rage. It's not the retyping I mind, it's the rethinking. I just can't face it.

The last time this happened to me, I didn't give up. WordStar was in an endless loop: beeping and inserting exclamation points in weird and what would otherwise be fascinating patterns. The keyboard was dead. Pulling out the printer interface card while the machine was running (not recommended!) stopped the malignant activity, but the keyboard was still dead. Eventually, there was nothing for it but CTRL-RESET. There went all my work...

But, wait. Maybe it's still in memory! I called up DDT and used the D command to examine all of memory, from the beginning. Sure enough, at hex 7849 on, there were my precious words! I used the DDT M command to move my text to 0100. Back to the system with CTRL-C. Save the recovered material (e.g., SAVE 6 HOPE). Not only can it be typed (TYPE HOPE), but I could use it as input to WordStar with no problems! (Well, there was garbage at the end, a small price to pay, and no doubt inserting a proper terminator with DDT would have cured that.)

Too late to recover that three-page letter I wrote a month ago - but I'm ready for next time. Don't wait until it happens to you. Practice now. Be prepared!

Keith K. Davison
239 Clinton Road
Brookline, MA 02146

More Circles and Such

We received several items amplifying the circle generation discussion begun by Daniel Lee in the May 1983 DDJ. In order to present them all together, this month the Letters column spills over to the sidebars on pages 14-16. - Ed.

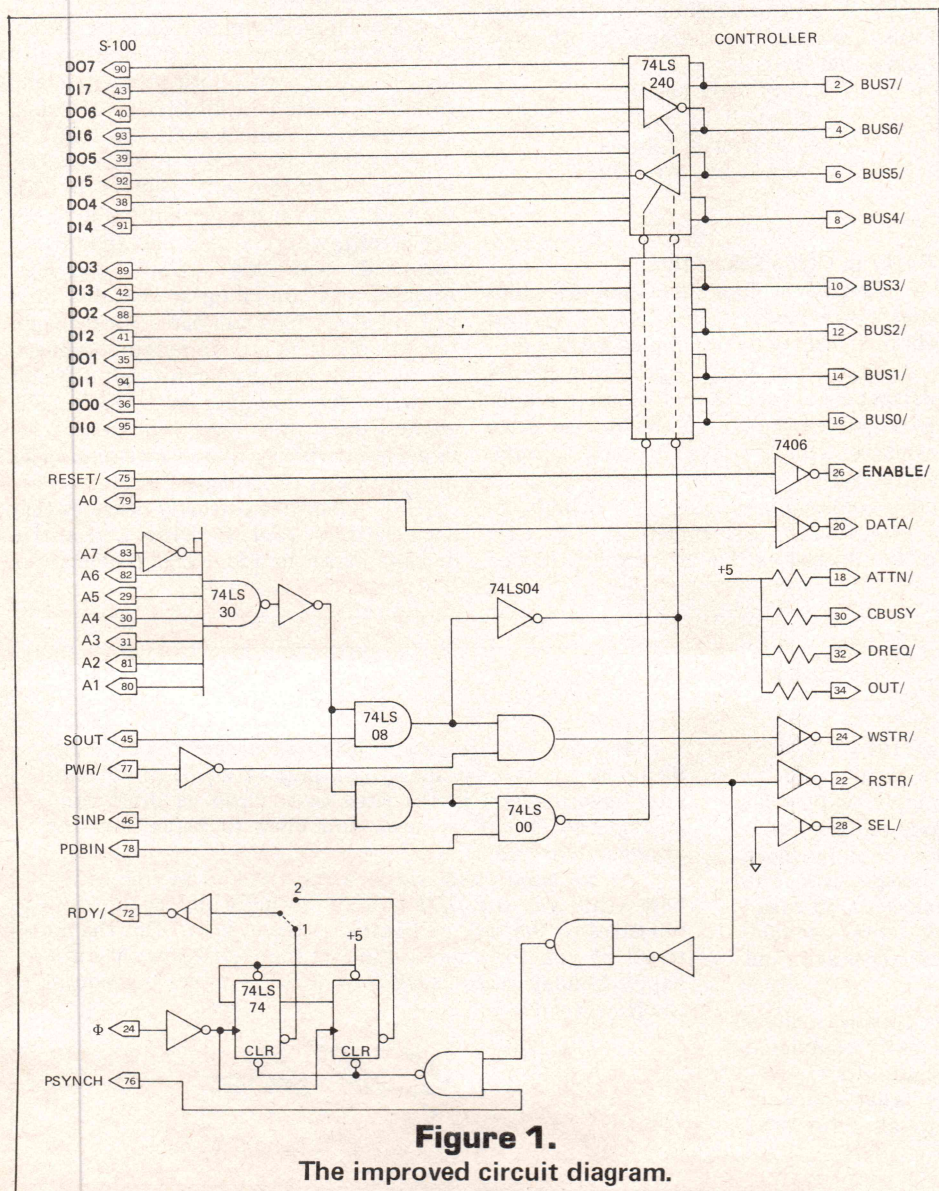


Figure 1.
The improved circuit diagram.

COHERENT™ IS SUPERIOR TO UNIX* AND IT'S AVAILABLE TODAY ON THE IBM PC.

Mark Williams Company hasn't just taken a mini-computer operating system, like UNIX, and ported it to the PC. We wrote COHERENT ourselves. We were able to bring UNIX capability to the PC with the PC in mind, making it the most efficient personal computer work station available at an unbelievable price.

For the first time you get a multi-user, multitasking operating system on your IBM PC. Because COHERENT is UNIX-compatible, UNIX software will run on the PC under COHERENT.

The software system includes a C-compiler and over 100 utilities, all for \$500. Similar environments cost thousands more.

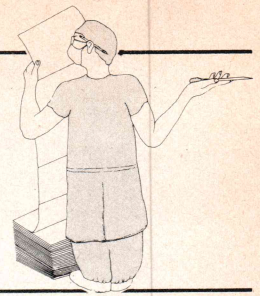
COHERENT on the IBM PC requires a hard disk and 256K memory. It's available on the IBM XT, and Tecmar, Davong and Corvus hard disks.

Available now. For additional information, call or write,

Mark Williams Company
1430 West Wrightwood, Chicago, Illinois 60614
312/472-6659



COHERENT is a trade mark of Mark Williams Company.
*UNIX is a trade mark of Bell Laboratories.



by D. E. Cortesi, Resident Intern

Readers of this column should know about two recent startups in the column industry. The first is a big-time effort that, we think, faces grave difficulties. The second promises to be a gem of great value and deserves to be widely read.

Big-Time Columnizing

Many of you will already know that Douglas Hofstadter has retired his "Metamagical Themas" in *Scientific American* and that Brian Hayes has stepped into that hallowed space (originally occupied by Martin Gardner's "Mathematical Games") with "Computer Recreations." His first effort, a consideration of the fundamental shift in thinking encouraged by spreadsheet calculators, appeared in the October issue. It's a welcome sign when the august *SciAm* recognizes the prevalence, and the fun, of personal computing, and we wish Hayes the best of luck.

System-Dependent Prose

However, columnizing isn't easy in the best of circumstances, and he faces some difficulties. We wonder how (given the magazine's huge circulation, the diversity of its readers' interests, and the diversity of the computers they use) he is going to proceed. Spreadsheets are safe enough for a first outing; everybody has one. But when it comes time to present algorithms, what language will he use? When it is time to draw a graph, can he avoid hardware dependency?

The infernal, niggling differences between one language (machine, keyboard, disk drive, etc. *ad nauseum*) are the curse of computer publishing, and nobody has found a decent way to pro-

duce system-independent prose. It isn't a new problem, either. The Association for Computing Machinery (ACM) faced it a long time ago and decided to publish algorithms in "presentation ALGOL," the ALGOL syntax augmented with a complicated set of typographical conventions. Donald Knuth faced it and opted to design his own CPU, MIX, and to present algorithms in its assembly language, MIXAL. These solutions are essentially the same. They proceed from the assumptions that (1) no two readers use quite the same language, and (2) each reader knows some language and system very well. Therefore all readers will be presented with a language that *nobody* knows and will all have to learn it; afterward they can translate to their own environment.

The personal-computer press usually operates on the theory that everybody knows and uses some form of BASIC, and that all readers are good enough programmers to convert from one BASIC's idiosyncracies to those of their own machine's BASIC. It isn't true, of course. We once published a program for the IBM PC (in another magazine) and later received a long distance call from an unhappy reader who simply couldn't understand why the program didn't work on his Victor 9000.

Another solution is to embrace the problem and publish system-dependent material only; hence we have magazines for the Apple, the Atari, the Color Computer, the IBM PC; for CP/M; for the Heathkit product line; for machines that use the 6502. And we have books that appear in multiple editions with only trifling changes in their content, like the

series of *The Power of XXXXCalc* books. This isn't a solution at all, of course; it only hides the problem. It drives up costs, fragments the audiences, and prevents the dissemination of good solutions to many users.

So how will Hayes handle the problem? Can he find a way to present problems and solutions to a mass audience in a way that will be interesting and useful to most of his readers? We'll see.

Bently's Pearls

Communications of the ACM, long-time flagship of computer science journals, launched a new column in its August issue. The columnist is Jon Bently, an Associate Professor at Carnegie-Mellon and author of *Writing Efficient Programs* (a book that, on the evidence of Bently's debut column, we certainly plan to buy).

The column's title is "Programming Pearls," the pearls being those parts of programs whose origination require "insight and craftsmanship." The pearls presented in the first column were delightful and gave us several hours of vigorous, enjoyable mental exercise. Bently has avoided the problem of system dependency by keeping his problems abstract and by presenting algorithms in English and, where necessary, in a simple pseudo-code.

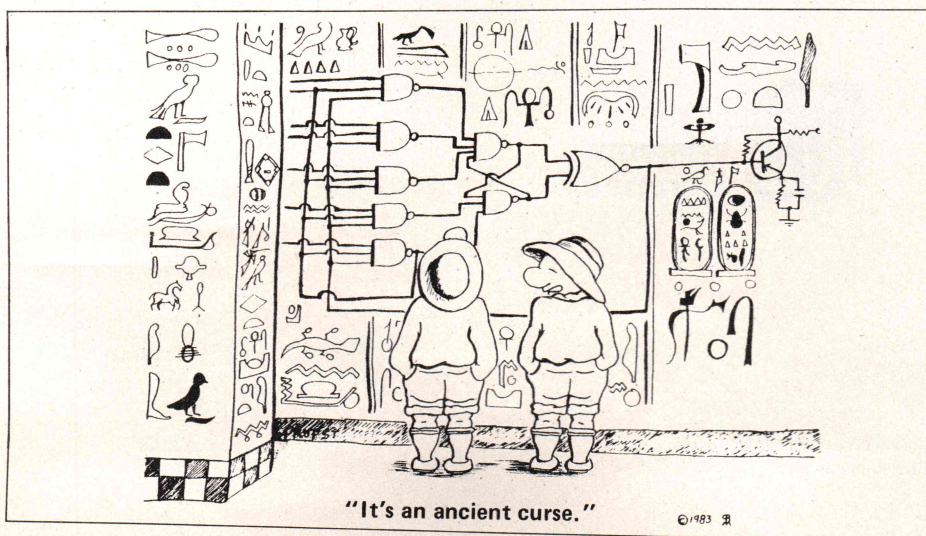
Bently plans to end each column with questions and problems, but he isn't asking for solutions from his readers (from our experience, we'd call that a mistake; the massed mental power of a motivated readership is awesome). Here is one of the problems he presented.

"Rotate a vector of N elements by I positions. For instance, with $N=8$ and $I=3$, the vector $ABCDEFGH$ is rotated to be $DEFGHABC$. It is easy to do this rotation in N steps if we have available an N -element intermediate vector. Can you rotate the vector in time proportional to N using only a few extra words of storage?"

We could; we ended up using five scratch variables. Furthermore, when we expressed our solution as a BASIC subroutine (taking variables I and N and vector V as input, modifying the contents of V), we found that with a single added line of code we could accept a negative value of I to mean "rotate right." Can you find a similar pearl for this oyster of Bently's? If you enjoy searching for that one, be sure to read the rest of his column. ■■■

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 191.



WHEN YOU WORK WITH YOUR MICRO FROM THE INSIDE, YOU CAN KEEP ALL THE BUGS ON THE OUTSIDE.

The only way to maximize your micro's potential—and keep your programs running smoothly while you experiment with new techniques—is to have a good advance idea of what will work, what won't, and why. Wiley's answer books reveal your micro's inner workings so you can get inside and understand those crucial concepts. Written with clarity and unusual detail by people who are tops in their fields, these are the guidebooks you'll reach for when you sit down to think at the screen.

THE UNIX™ OPERATING SYSTEM BOOK

Mike Banahan and Andy Rutter

THE UNIX OPERATING SYSTEM BOOK is a highly readable introduction to the features, capabilities, and limitations of this increasingly popular system—the brainchild of Bell Laboratories—that brought manageability and flexibility to previously unwieldy operating systems. The guide provides in-depth coverage of the UNIX editor, the “C” programming language, and text processing. You'll also find clear coverage of logging in, using the commands, “C” libraries, the filestore, the process environment, library routines, and system calls.

224 pages (1-89676-4) \$16.95

WORDSTAR™ AND CP/M® MADE EASY

John D. Lee

Wordstar is the most popular word-processing software for micros—and this complete guide introduces all its functions and features in an easy-to-understand, hands-on format. In just days, beginners learn basic writing, editing, on-screen formatting, and printing. More advanced users hone their skills with new strategies, “tricks,” and shortcuts for formatting, moving, and deleting text, creating and merging files. Special features include Mail/Merge,™ SpellStar,™ and the CP/M® Operating System.

244 pages (1-90188-1) \$17.95

CP/M® FOR THE IBM USING CP/M-86®

Judi N. Fernandez and Ruth Ashley

CP/M-86 is the widely accepted operating system used with the new IBM Personal Computer. When you fully grasp how the entire CP/M-86 package works, you're able to perform both routine functions and sophisticated tasks with maximum flexibility and efficiency. And this is the only self-teaching guide, ideal for both novices and experienced pros, that will enable you to effortlessly master the entire operating system. From the simplest tasks to the most complex, you'll learn through progressive steps and hands-on applications.

288 pages (1-89719-1) \$14.95

THE 8087 PRIMER

John F. Palmer and Stephen P. Morse

The 8087 from Intel is a chip with special capabilities for rapid numerical processing and computation that can be used in the IBM PC and machines from Zilog and Motorola. It's in demand everywhere, but no one understands the potential of the 8087 better than its designer: author John Palmer. From their professional perspective, Palmer and Stephen Morse (designer of the 8086) explain in clear detail how to use and program the 8087, how to interface it with 8086-based machines, and how to program it in various languages, including Fortran and Pascal.

224 pages (1-87569-4) \$16.95

THE UNIX™ OPERATING SYSTEM

Kaare Christian

UNIX runs on a wide variety of computers and has a broad range of applications to business, science and industry, making it one of the most important systems to have debuted in the past decade. This book is a readable, step-by-step introduction to the features and uses of this system, which was first developed at Bell Labs. It includes an abridged UNIX manual, with citations for forty of the most commonly used UNIX commands.

400 pages (1-87542-2) \$24.95

PC DOS

USING THE IBM PC OPERATING SYSTEM

Ruth Ashley and Judi N. Fernandez

Most manuals tell you *what* commands to use, but only this self-teaching guide explains *all aspects* of a command and leads you in hands-on practice to reinforce each new skill you acquire. Starting with the most basic task—creating a new file name—and progressing through thoughtfully graduated steps, you will master the entire PC DOS 1.1 system. Numerous examples clarify important new concepts, as you learn to create and erase files, copy files, run other programs, print data from files, and implement every other function of PC DOS.

288 pages (1-89718-3) \$14.95

More than two million people have learned to program, use, and enjoy microcomputers with Wiley microcomputer paperbacks. Look for them at your favorite bookshop or computer store!

FOR FASTER SERVICE

CALL TOLL FREE 1800 526-5368

In New Jersey, call collect (201) 342-6707

Order code #4-1432

VISA, MasterCard, American Express accepted on phone orders.



John Wiley & Sons, Inc.

605 Third Avenue, New York, N.Y. 10158

In Canada: 22 Worcester Road, Rexdale, Ontario M9W 1L1

Prices subject to change and higher in Canada.

UNIX™ is a trademark of Bell Laboratories.

CP/M® and CP/M-86® are registered trademarks of Digital Research.

Wordstar,™ MailMerge,™ SpellStar™ and DataStar™ are all trademarks of the MicroPro International Corporation.

UNLOCKING THE POWER OF COMPUTING

4-1432

C/UNIX PROGRAMMER'S NOTEBOOK

by Anthony Skjellum

In the last column, several references to the book *The C Programming Language* and its authors were made. Through my error, Brian Kernighan's name was misspelled consistently throughout the column. I'm sure that many readers noticed this immediately. Unfortunately, I didn't until I saw the column in print. [*We obviously missed it too. Our apologies. -Ed.*]

In this second Programmer's Notebook, I'll discuss how Unix-type environments can lead to non-interactive, and user-unfriendly, software. This is based on experiences I've had with several Unix and Unix-like systems running both standard and Berkeley Unix.

Skimming through the ads of the October *DDJ*, I noticed a very interesting item: the Computer Innovations ad concerning the soon-to-be-released version of their C compiler. As an option, this compiler will produce programs/data exceeding the previous limit of 64K segments imposed by all 8086 compilers (C or otherwise) that I've seen. Look for a review in this column early next year.

Unix Software

The Unix operating system was designed to reduce repetition of programming effort by permitting modular programs to be combined via pipes and tees. Since input and output are redirectable under Unix, simple programs could use console input and output for one application and be used as part of a pipeline for another. Thus, unmodified programs could be reharnessed for new applications to an extent not possible with previous operating systems.

Pipes and input-output redirection are two of the best and most well-known features of the Unix system. Microcomputer users have been very interested in adding these capabilities to their own operating environments. In the 8-bit world, this has been done chiefly through special subroutine libraries such as "The Unica," or in C runtime packages. For MS-DOS 2.0 users, the features are built into the operating system.

Despite the undisputed usefulness of pipelines and input-output redirection, their presence in Unix has lead to a serious drawback in the system's environment. This drawback is the proclivity to avoid interactive programs and to produce user-unfriendly software. Furthermore, the standard Unix console interface is weaker than under other operating systems. In the remainder of this column I will discuss

1. Terse (hard-to-remember) program names.
2. Lack of program sign-on and sign-off messages.
3. Lack of interactive mode to alleviate the need to re-execute a program several times to complete a set of operations.
4. Inconsistent use of switch (dash options) for controlling the specifics of program execution.
5. Lack of descriptive error messages.
6. Cryptic, incomplete, and erroneous documentation.
7. Software bugs: undocumented and documented.
8. Cryptic (or missing) internal help features.
9. Poor console interface provided by Unix.
10. Lack of system for finding program names by the function required.

Table 1
Unix Software Problem Areas

```
$ fct <RETURN>          (activate the program with no arguments)
usage: fct -abc file1 .. fileN  (help message)
$                          (shell prompt)
```

Figure 1.
Cryptic fct Session

```
$ fct <RETURN>          (activate the program with no arguments)
fct (version x.yz) as of dd-mm-yy
usage:
    fct [dash options] file1 .. fileN
dash options:
    -a    perform function "a" on files specified
    -b    perform function "b" on files specified
    -c    perform function "c" on files specified
    -v    verify each step before proceeding
note: -a and -b are mutually exclusive; -c may be used in conjunction
      with -b only.
End of execution.
$                          (shell prompt)
```

Figure 2.
Friendly fct Session

these weaknesses as I perceive them.

Non-Interactive Software

Because of the availability of pipes and input-output redirection, many Unix programs are designed to act as filters. Filters are programs which require a single sequential input data stream and produce a single output data stream. Such programs are suitable as pipe-fittings. Because of the way they handle data, they don't normally expect to be used interactively. In most cases this doesn't pose a problem for users. However, because such programs do not expect to deal directly with humans, but only with input and output streams, they can often be very unfriendly in handling errors.

The problem in the Unix operating system is that the same terse philosophy applied to filters also pervades most of the software available. This includes programs which are normally executed sequentially by the user from the console. The problems come in several areas and some of these problem areas are listed in Table I (page 14). The following paragraphs will elaborate each of the points listed in that table.

Terse Names

Unix program names are usually two or three letters long and tend to be cryptic. While this saves typing for experienced users, it's frustrating for new and occasional system users. Also, since the Unix system lacks an on-line indexing system for finding program names by function, it's not easy to find the right program based on the desired function alone.

Sign-ons and Sign-offs

Sign-on and sign-off messages are a common courtesy in the computer world. Virtually all standard Unix programs lack these two simple features. While this is understandable for filters, it is completely unnecessary for other programs. For example, if two versions of a program exist on a system, the only easy way to distinguish them is by their sign-on messages.

Besides sign-ons/sign-offs, it is also nice for a program to give progress reports during execution. This lets the user know how things are proceeding. Standard Unix software doesn't normally include such a feature.

Internal Help Features

Many programs include a feature summary option to help occasional and new users remind themselves of program operation. Many Unix programs also have this capability, but they are often extremely cryptic and include few English words to supplement the sample command line which they display. Figure 1 (page 14) displays a sample session in which a fictitious program, *fc*, is executed

from the shell with no arguments. The program responds with a cryptic help summary typical of actual Unix commands. In Figure 2 (page 14) the same *fc* program session is presented, but this time the program has been designed to provide a user-friendly help feature (and also to sign on and off).

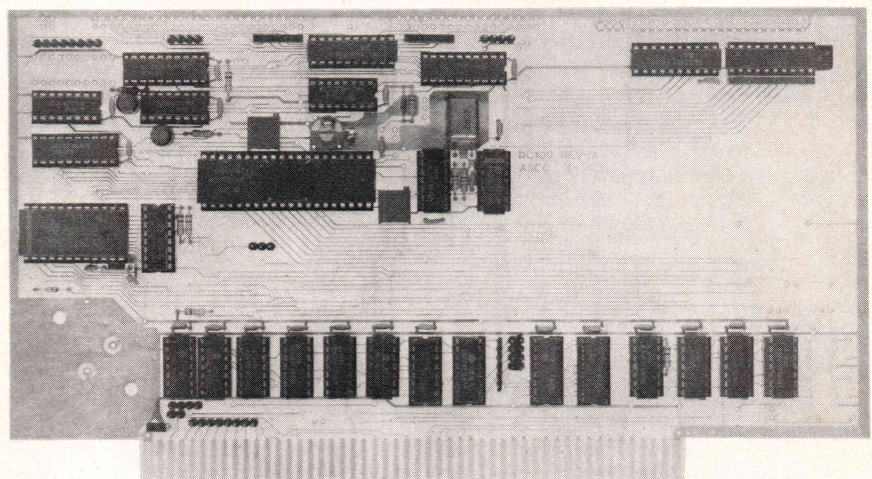
Interactive Modes

Interactive program modes provide a friendly environment for the user. When a program is used often, it may be executed several times consecutively. An interactive mode eliminates the need for consecutive execution since the user can enter all the

commands in one interactive session. This avoids unnecessary user effort and is probably more efficient from a system standpoint.

One reason that the interactive modes are missing is the lack of support for expanding Unix wildcard filenames from within a program. I find this limitation rather arbitrary, and I have written a program which solves the problem (see "Expanding Unix Wildcards," *DDJ*, No. 73, November 1982). The lack of direct support for such a function indicates that the whole Unix philosophy is geared towards non-interactive software tools.

FLOPPY DISK CONTROLLER / CALENDER CLOCK WITH HARD DISK INTERFACE



ASCC DC-100

- S-100 compatible.
- Will control up to four 8" and four 5¼" floppy disk drives in any combination.
- Uses latest Western Digital 2793 disk controller with integral PLL data separator.
- Separate register for 5" or 8", side select, single or double density.
- Host interface for Western Digital WD100X series hard disk controllers.
- Disk controllers (hard & floppy), clock may be programmed for vectored interrupts.
- MM 58167AN calendar clock chip w/160 mah lithium battery backup.
- Software drives included.
- Assembled, tested and burned-in for maximum reliability.
- 1 year warranty.

PRICING . . .

DC-100	\$269.00
WD-1002-HDO (if purchased w/DC-100)	295.00
UPS shipping	5.00
Calif. residents add tax	

Dealer and OEM inquiries invited

Delivery Stock to 6 weeks.

COMING SOON . . .

AN RGB color video/serial/parallel board	\$349.00
64K non-volatile static ram	349.00
2 — polyphonic music boards	N/A
Z80B CPU / 2-serial	N/A
2650A-1 CPU / 2-serial / parallel / 2M byte map	349.00

ASCC

BOX 40, ALPINE, CA 92001
(619) 445-9071

AMERICAN SCIENTIFIC COMPUTER CORP.
A California Corporation

Circle no. 3 on reader service card.

“ I have been using
TransForm for all
 kinds of programming
 ... my time has been cut in half...
 this is one of the very few utilities
 I have never regretted
 purchasing for one
 instant.”

by Dr. J.C. Gilland
 Graphion/Vortex

Whether you are just a beginner, someone reasonably familiar with programming, or a veteran, **TransForm** can help you.

What is TransForm?

TransForm is a better way to write programs in Microsoft BASIC®. It is a structured translator which does away with the constraints of line numbers and the single line IF. It adds the high level structures of the block IF-ELSEIF-ELSE-ENDIF, REPEAT-UNTIL, and ON-GOTO-ENDGOTO—similar to languages like Pascal. Symbolic labels are used (such as “GOSUB @ QUICKSORT”). A very versatile INCLUDE-DECLARE feature has been added to enable you to build a library of the subroutine modules you use every day—write it once, then INCLUDE it. High level, structured code in—ordinary BASIC out. It's that quick and easy.

Now! Three great ways to order TransForm:
The TransForm Tutor. Wanting to learn how to write clean, reliable, structured programs? Let the Tutor teach you with its detailed, easy to learn, step-by-step examples and tutorial. **Only \$39.95***

The TransForm Standard. Already familiar with structured programming? The Standard begins with a more complex example (a utility program) so you can dive right in. **Only \$39.95***

The TransForm Expert. Writing complex program series as a professional? The Expert provides the additional utilities of a source code formatter, destination code optimizer (for use with BASCOM® 5.3), and a “REL” file disassembler. Everything you need to master your applications. **Only \$39.95***

Place your order for **the Tutor, the Standard, or the Expert** toll free now: **1-800-845-7055**. Direct technical questions to 1-803-244-8174. VISA, MC, COD, company PO, or check accepted. For CP/M® *Introductory price.

From the same company that created MCDISPLAY, the powerful display interface for Microsoft BASIC which allows you to develop screen-oriented application packages.

MCI Master Computing Inc.

**“Practical Solutions Through Technology”
 A Division of EDI Technology Companies
 11 Regency Hills Drive, Greenville, SC 29615**

Command-line Switches

Unix programs use command-line switches (dash options) to specify the particulars of program execution. Switches usually consist of a dash character (“-”) followed by a single letter. Unfortunately, Unix programs use these switches inconsistently. For example, the **rm** command allows only limited positioning of switches, permitting them only before filenames on the command line. Some programs allow several switches to be combined after a single dash character (e.g., **-abcdef**). Others use the plus (“+”) character to activate a program feature and a dash to deactivate it. Generally, it is difficult to remember all the different possibilities, limitations, and defaults imposed by the various programs.

The difficulties with command-line switches under Unix led me to write the ARGUM package which was published in the August 1982 issue of *DDJ* (No. 70). That article proposed a program which would handle switches in a consistent way. Existing Unix programs could be changed to use ARGUM, thus eliminating one degree of inconsistency from the operating environment. Alternatively, the less powerful Unix III **getopts()** facility could be used.

Lack of Descriptive Error Messages

The lack of descriptive error messages is a real problem, especially for inexperienced users. One offender is the **eqn/troff** system used for equation and text phototypesetting. These programs report errors as “Syntax Error” between two line numbers. They don't echo the erroneous text or equation, and the line numbers aren't always useful because header files change the length of the source text.

Another problem stems from the way Unix reports failures. Programs which attempt to open a file and fail get an error code. They subsequently report the failure as “cannot open file.” While this is correct, it doesn't tell the user if the file is nonexistent or if a file protection violation has occurred. Similarly, when users try to change their current directory to one for which they have no read privileges, a “bad directory” message is displayed by the Unix shell.

Documentation

Documentation is a real problem in the Unix system. Most programs are documented in a standard form which is typically very terse. The examples provided are often very complicated and don't clearly illustrate how to use the program for simple purposes. Because of the poor documentation, some casual users think of Unix systems as secret societies since only the indoctrinated can tell them how the system and programs work.

Another problem with documentation is that it sometimes doesn't reflect the current state of a program. Most notable on the system I use are undocumented features of the standard Unix editor `ed`. The standard form provides no line edit command ("`x`") while the installed version does. Yet, the documentation does not explain this.

Unix documentation is clearly the weak link in the system. It makes some of the mysterious concepts of the system seem impossible to grasp and reduces productivity through its terseness.

Software Bugs

Another problem which makes a Unix system user-unfriendly (actually user-hostile) is the presence of undocumented bugs in important software packages. For example, bugs exist in the `eqn/troff` system. These can be circumvented, but the methods are known only to a few experts. No generally available documentation exists for avoiding such problems.

Since the Unix system comes with source code, it should be feasible for individual users to change system software to their specific needs. Unfortunately, the

source code which accompanies Unix is mostly comment-free and is therefore difficult to understand without significant effort.

User Console Interface

The user console interface is not extremely good under Unix. For example, when a character is deleted, Unix does not actually remove the character typed but just moves the cursor back one space. There is no standard mechanism for having a line retyped, and backspacing after typing a tab doesn't produce the correct results. Furthermore, when a control character is typed, it is displayed in the form `^CHAR`, but when deleted, the cursor moves back only one space (leaving the caret).

The weak user interface points again to the philosophy that most software will not be interactive. However, the standard Unix editor makes no attempt to improve the interface for the purpose of interactive editing. It is so unfriendly that most users resort to other editors (usually screen editors).

Programmers can only provide their programs with a superior console interface by using the raw terminal mode. Un-

fortunately, this mode is more expensive in terms of input-output cost. For user-friendly, screen-oriented software, it's the only way to go.

Conclusion

Unix is a powerful operating system, and is certainly one that I enjoy using. It does, however, have a number of user-unfriendly aspects, and the system philosophy has led to a predilection for non-interactive software.

I look forward to hearing how other Unix users perceive the Unix environment.

DDJ

Reader Ballot

Vote for your favorite feature/article
Circle Reader Service No. 193.

YOURDON PRESS has tools for technicians and tools for managers.

C NOTES: A Guide to the C Programming Language by C.T. Zahn
This well-organized reference manual focuses on the high-level language C, which supports structured program design, allows separately compiled modules, and contains important features for ease of program maintenance. The text provides a perspective on the balance inherent in C between purity and practicality and between portability and efficiency. The book describes the language syntax, external and local declarations, and macro preprocessor facilities, among other topics. Numerous examples illustrate the concepts. 112 pages; \$21.25.

CONTROLLING SOFTWARE PROJECTS: Management, Measurement & Estimation, by Tom DeMarco with a Foreword by Barry W. Boehm. The failure of software projects is commonplace and typically quantitative in nature: overruns of cost, time, and effort. Managers need a system to make effective project management less elusive. Tom DeMarco shows managers how to make rational, metric-based estimates, how to track progress, and how to detect potential problems before they lead to costly overruns. 150 diagrams; sample set of models; statistics primer; sample program to compute code volume. 296 pages; \$29.75.

TO ORDER: Residents of CA, NY, and WA states, please add applicable sales tax. Prices include \$1.25 for shipping and handling; Or charge on ☐ VISA or ☐ MASTERCARD.
Card No. _____
Exp. Date _____
Signature _____
☐ Enclosed is \$21.25 for C Notes.
☐ Enclosed is \$29.75 for
Controlling Software Projects.
☐ Please send me your latest catalog.
Name _____
Address _____
City, State, Zip _____
Telephone() _____

DDJ 12/83

Yourdon Press 1133 Ave. of Americas
New York, NY 10036-6748

Yourdon Press 1133 Avenue of the Americas
New York, NY 10036-6748 (212) 391-2828

Circle no. 132 on reader service card.

Faster Circles for Apples

Daniel Lee's article, "Fast Circle Routine," in *DDJ* No. 79 (May 1983) inspired me to create a similar circle-making approach for the Apple. Although I first wrote an Applesoft BASIC program to test the logic, I chose variable names that could serve for both BASIC and assembly language. Thus "X" and "Y" refer to the X and Y registers of the Apple's 6502 chip, and "Xcrd" and "Ycrd" are the coordinates for the points of the circle.

A circle of radius R centered at (Xmid, Ymid) is described by the familiar formula

by Myron L. Pulier

Myron L. Pulier, M.D., 101 Cedar Lane, Teaneck, New Jersey 07666.

$$(Ycrd - Ymid)^2 + (Xcrd - Xmid)^2 = R^2$$

where Xcrd is the horizontal variable, Ycrd is the vertical variable, and Xmid and Ymid are constants. Differentiating with respect to Xcrd gives

$$2 * (Ycrd - Ymid) * dYcrd / dXcrd + 2 * (Xcrd - Xmid) = 0$$

whence

$$dYcrd / dXcrd = -(Xmid - Xcrd) / (Ymid - Ycrd)$$

The last equation implies that, in drawing the circle, if we increase Xcrd by 1 to plot the next point we must decrease Ycrd by

$$(Xmid - Xcrd) / (Ymid - Ycrd)$$

The slowest operation here is division by Ymid - Ycrd, which must be performed each time we want a new value for Ycrd.

We can reduce the number of these divisions by evaluating the expression for only one eighth of the circle and by plotting the rest of the circle symmetrically about the coordinate axes and about a diagonal.

It is best to select the upper left extreme of the circle as the starting point. According to the Apple coordinate system, where point (0,0) is the upper left corner of the screen, our starting point is given by

$$(Xmid - R / \text{SQR}(2), Ymid - R / \text{SQR}(2))$$

From here we move to the right and stop at the extreme top of the circle, which is point (Xmid, Ymid - R). This choice of starting and ending points facilitates a simple FOR-NEXT program loop (FOR Xcrd = Xmid - R / SQR(2) to Xmid) and avoids the divide-by-zero error we might encounter at the extreme right and left of the circle, where the slope is undefined.

Reader Commentary

More Fast Circles . . .

Dear *DDJ*,

Daniel L. Lee's algorithm has got to be faster than Microsoft's pedestrian CIRCLE command, but both suffer from the same malady: they reinvent the wheel — only this time it's square!

When I think I've discovered a marvelous algorithm, I wonder if I've outsmarted the professionals. I usually haven't. But hope springs eternal. I search the literature anyway. My brainchild is at least 17 years old [B. K. P. Horn, "Circle Generators for Display Devices," *Computer Graphics and Image Processing* (5), pp. 280-288 (1976)].

Neither trigonometry nor calculus is needed to devise a circle generator. For a circle of radius R one wants to plot points (X,Y) with integer coordinates which most nearly solve the equation

$$X^2 + Y^2 = R^2$$

The difference between the left side and R^2 is a measure of nearness. A suitable circle generator simply chooses successive points to minimize this difference. The enclosed listing (see Listing Three, page 30) is a rendering of such an algorithm. It generates points for about one eighth of the

circle and, using the symmetry of the circle, plots eight points for each point generated. For use with digital plotters, the algorithm is invoked eight times forward and backward so that the points are of concentric circles; the low algorithm is plotted in counter-clockwise order. I enclose a plot of concentric circles in low resolution to

exhibit the algorithm's behavior (see Figure 2, below).

William A. McWorter, Jr.
Mathematics Department
Ohio State University
231 W. 18th Avenue
Columbus, OH 43210

(Listing Three begins on page 30)

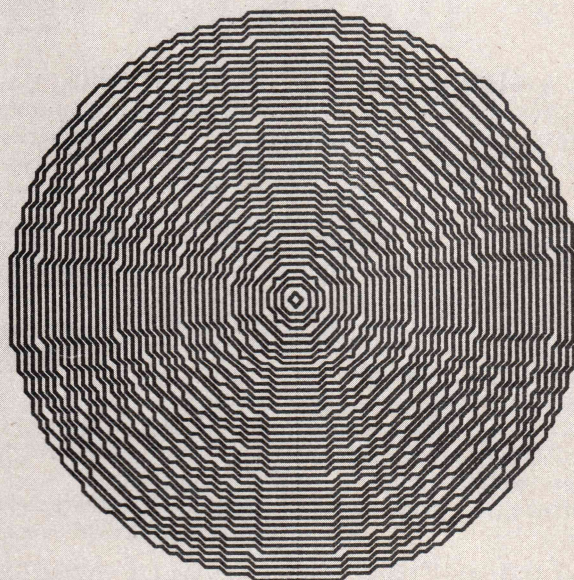


Figure 2.

See Figure 1 (at right).

For each point that we find by the above method, we can generate seven symmetric points by reflection through the horizontal and vertical diameters of the circle and by exchanging Xcrd with Ycrd. A point (Xcrd,Ycrd) is $\text{ABS}(X_{\text{crd}} - X_{\text{mid}})$ distant from the vertical axis of the circle. Since its reflection should be the same distance from this axis, its abscissa is $2 * X_{\text{mid}} - X_{\text{crd}}$. Similarly, reflection through the horizontal diameter gives an ordinate of $2 * Y_{\text{mid}} - Y_{\text{crd}}$. To reflect a point through the diagonal line that runs from upper left to lower right, we find the point whose distance to the vertical axis equals the distance of the original point to the horizontal axis and whose distance to the horizontal axis is the same as the original point's distance from the vertical axis, namely

$$(X_{\text{mid}} + Y_{\text{mid}} - Y_{\text{crd}}, X_{\text{mid}} + Y_{\text{mid}} - X_{\text{crd}})$$

This new point can now be reflected as before through the vertical and horizontal axes.

In Listing One (page 21), subroutine 9000 plots four points symmetrically about the horizontal and vertical axes. Lines 10050 and 10055 switch the X and Y coordinates, then line 10060 calls 9000 to plot the four new points. The param-

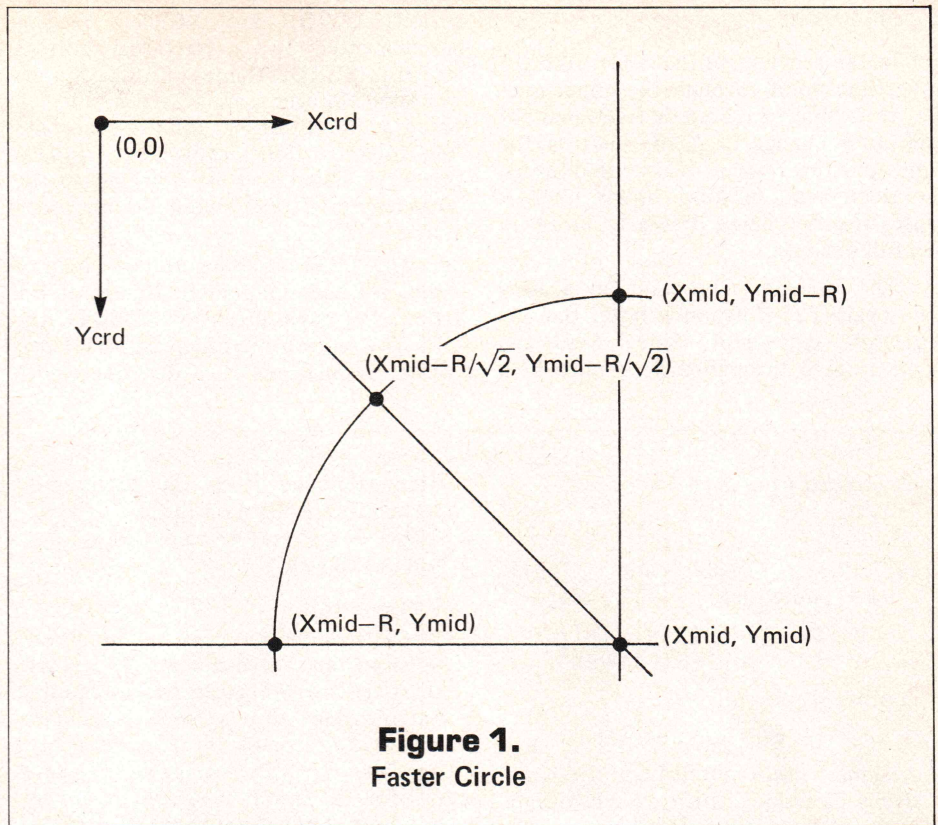


Figure 1.
Faster Circle

Reader Commentary

... And Fast Ellipses

Dear Sirs:

Mr. Daniel Lee, in the May '83 issue, presented a fast circle generator. It compensated for any given screen aspect ratio, and as such may be used as an ellipse generator. I submit the algorithm described below as an even faster alternative. The speed improvement results from the elimination of all division and most of the multiplication. The approach taken could easily be modified to allow the generation of arcs.

The method which I present here is based on the equation of the circle, and a trick which eliminates a great deal of multiplication. There is no calculus or trigonometry involved, implicitly or explicitly.

The equation of the circle is well known:

$$x^2 + y^2 = r^2 \quad [1]$$

where r is the radius. Since we want to minimize multiplication, we have to use "magic." A magical property of the positive integers is that the square of a positive integer n is the sum of the first n odd numbers. This means that if we want to compute x^2 for each x we can actually plot (i.e., each integer x), we only need to know which odd

numbers to add up. The same applies to y^2 .

In order to plot a circle, we might start at the point $(0, r)$ and plot towards $(r, 0)$, using symmetry to generate the other arcs of the circle. This would mean that x would go from 0 to r , y would go from r to 0, x^2 would go from 0 to r^2 , and y^2 would go from r^2 to 0. It is easier than it first appears to calculate y^2 . Note that y^2 is the sum of the odd numbers from 1 to $2y-1$. In the initialization phase it will be necessary (perhaps) to compute y^2 directly, but for $y' = y-1$, $y'^2 = y^2 - (2y-1)$.

Above I said "perhaps" because it develops that one does not need to refer directly to y^2 or even to x^2 . The procedure for drawing the circle requires that we assume, as we did above, that we will draw primarily from $(r, 0)$ to $(0, r)$ and use symmetry to generate the rest of the points. As we compute the points for the primary arc, we maintain a total e . The total starts at 0; for every time we actually move in the positive x direction, we add $2x-1$ to e ; for every time we actually move in the negative y direction, we subtract $2y-1$ from e . We decide precisely which step or combination of steps to take by insisting that the e that would result from the step or combination of steps be as close to 0 as possible.

An Ellipse

To generate an ellipse is a slightly more complex matter, but in the end we lose little speed. The equation for an ellipse centered at the origin is

$$b^2 x^2 + a^2 y^2 = a^2 b^2 \quad [2]$$

where b is the positive y -intercept, a is the positive x -intercept, and a/b is the resulting aspect ratio. I claim that in order to successfully trace the ellipse we need only do exactly as we do for the circle, but we must multiply every reference to x by b^2 and every reference to y by a^2 . In other words, every time we actually move in the positive x direction, we add $b^2(2x-1)$ to e ; for every time we actually move in the negative y direction, we must subtract $a^2(2y-1)$ from e . Again we decide which step or combination of steps to take by insisting that the e that would result from the step or combination of steps be as close to 0 as possible. In this case we are plotting from $(0, b)$ to $(a, 0)$.

If perhaps the terms $b^2(2x-1)$ and $a^2(2y-1)$ look like they involve too much multiplication, please realize that in fact no multiplication is required. For example, we would already know the evaluation of $b^2(2x-1)$ to

(Continued in box on page 20)

ters R2, X2, Y2, and XY have been introduced to speed computation.

In the segment of the circle from the upper left point through the upper middle, the change in Ycrd is fractional for each unit change in Xcrd. Because the Apple plotting routine deals with integers, the decrement in Ycrd builds until it causes the line being drawn to move up one full position.

The assembly program in Listing Two (page 22) runs much faster than its Applesoft equivalent. Since Xcrd can range from 0 through 279, it must be a

double-precision variable. It occupies locations XCRDH and XCRDL. Ycrd is supplemented by a fractional portion stored in YCRDF. Names of other double-precision integer parameters are terminated with -H or -L for the high- and low-order portions, respectively. Single-precision assignment is indicated in the comments by "<—", while double precision is "<<—".

The TEST program plots a circle of radius 40 and midpoint (120,80). It initializes the hires screen by calling TURN-ON. The subroutine called EIGHTH performs calculations for the one-eighth

circle. Here the first order of business is to approximate the value of $R/\text{SQR}(2)$ by using $R*3/4$ instead. Note that $3/4$ in decimal is $1/2 + 1/4$, or 0.11 in binary.

The next lines of the assembly program are a straightforward translation of their Applesoft equivalents. Lines 75 and 76 initialize the value of YCRDF to 0. PLOTFOUR is called in lines 104 and 119 to place four points symmetrically about the horizontal and vertical axes of the circle. PLOTFOUR uses the Applesoft HLOT routine to perform the actual plotting. HLOT requires that the horizontal coordinate be in the Y and X registers,

(Continued from page 19)

be, say, e_x . To determine e_x' when $x' = x+1$, note that

$$\begin{aligned} b^2(2x'-1) &= b^2[2(x+1)-1] \\ &= b^2(2x-1) + 2b^2; \end{aligned}$$

in other words,

$$e_x' = e_x + 2b^2.$$

A similar result obtains for the negative y direction, which we will simply state:

$$e_y' = e_y - 2a^2.$$

Algorithm Summary

To summarize the algorithm: start with the point (0,b). Initialize e to 0, e_x to b^2 , e_y to $2a^2b - a^2$, e_{xy} to $e_x + e_y$. Plot the current point and corresponding points in the other quadrants of the ellipse. Choose the next point so that e plus e_{whatever} is minimized. Set e according to that choice, and update e_x , e_y , and e_{xy} . When the point (a,0) is arrived at, the ellipse is complete.

The Listing

The program shown in Listing Four (page 30) is an MBASIC program intended to interface to an LSI ADM-3A terminal. Obviously, if speed is a concern, BASIC is not the language of choice. I chose it to permit the program to be tried out basically anywhere, since my facilities for computer graphics are one-of-a-kind.

Lines 1050-1240 are the routine itself. The point-plotting routine is on lines 1310-1341.

Caveat

There is one thing that the implementor should be aware of before he or she starts, to prevent untraceable bugs. The formulae for e_x , e_y , and e_{xy} include squares of a and b . These squares accumulate to a large total rather quickly. The solution is to use a

wide word to store the total, and perhaps (depending on the size of your screen in pixels) the values of e_x , e_y , and e_{xy} as well.

Drawing Arcs

The method can be modified to draw arcs (see Figure 3, below) elliptical or otherwise, with careful initialization and a well-considered termination condition. The initialization involves calculating e_x , e_y , and e_{xy} for the initial point of the arc to be drawn. The routine should terminate when the last point of the arc is drawn. The actual coordinates of the final point should be calculated in some fashion that allows for rational numbers, and then a point with integer coordinates should be chosen that approximates the actual point. This can be done by using the equation of the ellipse. In other words, the best integer approximation (x_i, y_i) of the terminating point (x, y) is the one for which $(bx_i)^2 + (ay_i)^2$ is closest to

$(ab)^2$. Again, the integer coordinates of the final point should be computed in the initialization phase and used as the termination condition.

Conclusion

This routine can draw an ellipse quickly, using no multiplication once initialized. It should be easily implemented in 68000 assembly language, owing to that processor's 32-bit register operations. A little more difficulty should be anticipated by users of the 8086, 6809 or Z80, though their 16-bit addition capabilities can be used to advantage. HLLs can speedily draw circles with this routine, as well, because of its incremental nature. And finally, the algorithm can draw arcs easily.

Michael T. Enright
2360 Hosp Way, #132
Carlsbad, CA 92008

(Listing Four begins on page 30)

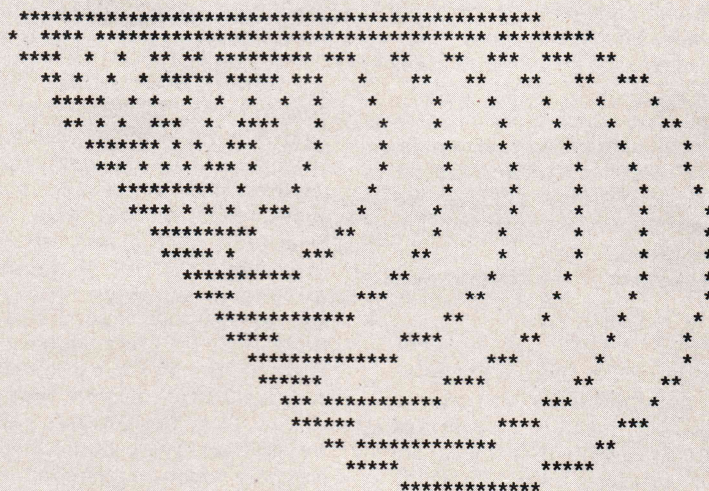


Figure 3.

and the vertical coordinate in the A register. HPlot returns these coordinate values at the zero-page locations named LASTHH, LASTHL, and LASTV.

The division sequence starts on line 135. Since the divisor and dividend are single precision, we can use a technique that divides a one-byte divisor stored in DIVISOR into a one-byte dividend held in the A register. The eight shift operations and eight subtractions required are counted via the X register. The result of the division is a binary fraction generated in QUOTIENT. This quotient is subtracted from the previous value of YCRDF. If a

borrow is required, we decrement the integer portion of Ycrd. In any case, Xcrd must be incremented by 1 in a double-precision operation. The FOR-NEXT loop of the BASIC version is implemented in assembly language by counting the value of R2 down through zero, since R/SQR(2) points will be plotted for one eighth of the circle.

Using zero-page locations for variables and parameters and a faster division algorithm will increase speed, but the bottleneck is the Applesoft HPlot routine, which maps horizontal and vertical coordinates into the Apple video locations.

Replacing that routine with a table lookup results in very fast circle generation.

With some modification the "faster circle" technique can produce filled-in disks or wedges for pie charts. It can also rotate and translate shapes and objects quickly for animation effects.

(Listings begin below)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 195.

Circles (Text begins on page 18)

Listing One

1000 *****TURNON

1005 Hgr
1010 Hcolor = 3

2000 *****TEST

2005 R = 40
2010 XMID = 120
2015 YMID = 80
2020 Gosub 10000
2025 End

9000 *****PLOTFOUR

9005 Hplot H,V
9010 Hplot X2 - H,V
9015 Hplot X2 - H,Y2 - V
9020 Hplot H,Y2 - V
9025 Return

10000 *****EIGHTH

10005 R2 = R * .7
10010 YCRD = YMID - R2
10015 X2 = XMID + XMID
10020 Y2 = YMID + YMID
10025 XY = XMID + YMID
10030 For XCRD = XMID - R2 To XMID
10035 H = XCRD
10040 V = YCRD
10045 Gosub 9000
* PLOT 4 POINTS

10050 V = XY - XCRD
10055 H = XY - YCRD
10060 Gosub 9000
* PLOT 4 POINTS

10065 YCRD = YCRD -
(XMID - XCRD) /
(YMID - YCRD)

10070 Next
10075 Return

END OF LISTING

PROGRAM LENGTH = 421 BYTES,
TOTAL OF 31 LINE NUMBERS

27 TOTAL NON-REM STATEMENTS,
6 TOTAL REMARKS

END
PR#0

End Listing One

(Listing Two begins on page 22)

Circles (Listing continued, text begins on page 18)

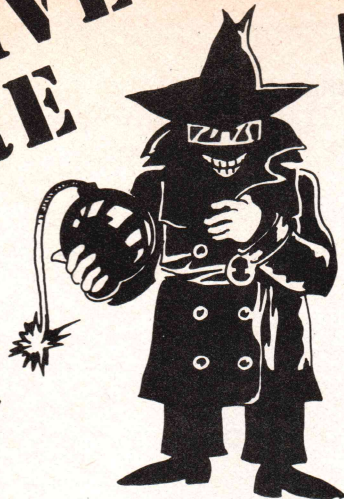
Listing Two

```
0800          3      LST
0800 2C 50 C0    4  TURNON  BIT GRAPHICS
0803 2C 52 C0    5          BIT FULL
0806 2C 54 C0    6          BIT PAGE1
0809 2C 57 C0    7          BIT HIRES
080C A9 FF       8          LDA #$FF
080E 85 E4       9          STA HCOLOR
0810 A9 20      10         LDA #$20
0812 85 E6      11         STA HPAGE
0814          12      ;
0814 A9 28      13  TEST    LDA #40          ;R<-40
0816 8D 2F 09   14          STA R
0819          15      ;
0819 A9 78      16          LDA #120        ;XMID<(-120
081B 8D 37 09   17          STA XMIDL
081E A9 00      18          LDA #0
0820 8D 36 09   19          STA XMIDH
0823          20      ;
0823 A9 50      21          LDA #80         ;YMID<-80
0825 8D 3C 09   22          STA YMID
0828          23      ;
0828 20 64 08   24          JSR EIGHTH      ;DRAW CIRCLE
082B 60         25          RTS
082C          26      ;
082C AC 2D 09   27  PLOTFOUR LDY HH          ;HPLOT H,V
082F AE 2E 09   28          LDX HL
0832 AD 31 09   29          LDA V
0835 20 57 F4   30          JSR HPLOT
0838 AD 33 09   31          LDA X2L        ;HPLOT X2-H,V
083B 38         32          SEC
083C E5 E0      33          SBC LASTHL
083E AA         34          TAX
083F AD 32 09   35          LDA X2H
0842 E5 E1      36          SBC LASTHH
0844 AB         37          TAY
0845 AD 31 09   38          LDA V
0848 20 57 F4   39          JSR HPLOT
084B A4 E1      40          LDY LASTHH      ;HPLOT H2-H, Y2-V
084D A6 E0      41          LDX LASTHL
084F AD 3B 09   42          LDA Y2L
0852 38         43          SEC
0853 E5 E2      44          SBC LASTV
0855 20 57 F4   45          JSR HPLOT
0858 AC 2D 09   46          LDY HH
085B AE 2E 09   47          LDX HL
085E A5 E2      48          LDA LASTV
0860 20 57 F4   49          JSR HPLOT
0863 60         50          RTS
0864          51      ;
0864 AD 2F 09   52  EIGHTH  LDA R          ;R2<-R*3/4
0867 4A         53          LSR
0868 18         54          CLC
0869 6D 2F 09   55          ADC R
```

(Continued on page 24)

SUBVERSIVE SOFTWARE

*So cheap and useful
it's...dangerous!*



A radical idea in software development.
Useful Pascal programs, with:

- Complete annotated source listings;
- Disk(s) with source code and compiled code;
- User manual;
- Complete programmer documentation describing data structures and algorithms; and giving suggestions for modification.

Modify them, include them in your own systems, or simply use them.

A growing library of software tools you'll find hard to resist.

SUBVERSIVE SOFTWARE

TPL

The Text Processing Language. A text-file runoff program consisting of a set of text-processing primitive commands from which more complex commands (macros) can be built (as in Logo). Features include:

- Complete customization of text processing through macro definition and expansion, looping structures, and conditional statements;
- Adapts to any printer;
- Pagination;
- Text justification and centering;
- Indexing and tables of contents;
- Superscripts and subscripts;
- Bolding and underlining;
- Multiple headers and footers;
- End notes and footnotes;
- Widow and orphan suppression;
- Floating tables and 'keeps.'

\$50

SUBVERSIVE SOFTWARE

DBX

Blocked Keyed Data Access Module. Maintains disk files of keyed data. Can be used for bibliographies, glossaries, multi-key data base construction, and many other applications.

- Variable-length keys;
- Variable-length data;
- Sequential access and rapid keyed access;
- Single disk access per operation (store, find, delete) in most cases;
- Multiple files;
- Dynamic memory allocation for RAM-resident index and current "page" of entries;
- Includes demonstration program and testbed program.

\$50

SUBVERSIVE SOFTWARE

PDMS

The Pascal Data Management System. A user-oriented data management system in which numeric and alphanumeric data are stored in tables with named columns and numbered rows. Currently being used for dozens of different kinds of business and scientific applications, from inventory management to laboratory data analysis. Includes over 20 Pascal programs; more than 10,000 lines of code. Main features include:

- Maximum of 32,767 rows per file;
- Maximum of 400 characters per row, and 40 columns per table;
- Full-screen editing of rows and columns, with scrolling, windowing, global search/replace, and other editing features;
- Sorting, copying, merging, and reducing routines;
- Mailing label program;
- Reporting program generates reports with control breaks, totals and subtotals, and selects rows by field value; many other reporting features;
- Cross-tabulation, correlations, and multiple regression;
- Video-display-handling module;
- Disk-file-handling module.

Many other features. UCSD formats only.

\$250

SUBVERSIVE SOFTWARE

ZED

Full-screen text editor; designed to be used either with TPL or by itself.

- Full cursor control;
- Insert mode with word wrap;
- 'Paint' mode;
- Single-keystroke or dual-keystroke commands;
- Command synonyms;
- Global search and replace;
- Block move, block copy, and block delete.

\$50

SUBVERSIVE SOFTWARE

SCINTILLA

A log logit curve fitting program for radio-immunologic data; must be used with PDMS (described above).

- Multiple protocol files;
- Quality control files;
- Four-parameter non-linear curve fit.

UCSD formats only. **\$250**

SUBVERSIVE SOFTWARE

CHROME

Chromatography data analysis program:

- Graphic display of analog data;
- Panning and zooming;
- Automatic peak-finding and baseline calculation;
- Full interactive peak editing;
- Computation of peak areas;
- Strip charts on C. Itoh and EPSON printers.

\$100

SUBVERSIVE SOFTWARE

PLANE

Planimetry program:

- Bit-pad entry of cross sections;
- Real-time turtlegraphics display;
- Calculation of areas;
- Saves calculations to text file.

\$100

SUBVERSIVE SOFTWARE

MINT

A terminal emulation program for communication between computers of any size.

- User-configurable uploading and downloading of files;
- X-ON/X-OFF and EOB/ACK protocols;
- Interrupt-driven serial input (for Prometheus Versacard in Apple II);
- Printer-logging.

\$50

For more information, call 919-942-1411. To order, use form below or call our toll-free number: 1-800-XPASCAL

Check appropriate boxes:

FORMAT

- ☐ 8" UCSD SSSD
☐ 5 1/4" Apple Pascal
☐ 5 1/4" UCSD IBM PC 320k
☐ 8" CP/M SSSD
☐ 5 1/4" IBM MS-DOS
☐ 5 1/4" CP/M Osborne

PRODUCT

- ☐ DBX
☐ PDMS
☐ TPL
☐ ZED
☐ MINT
☐ SCINTILLA
☐ CHROME
☐ PLANE

PRICE

- ☐ \$ 50
☐ \$250
☐ \$ 50
☐ \$ 50
☐ \$ 50
☐ \$250
☐ \$100
☐ \$100

Name _____

Address _____

☐ MasterCard

☐ VISA

☐ Check

☐ C.O.D.

(Please include card # and expiration date)



SUBVERSIVE SOFTWARE

A division of Pascal & Associates,

135 East Rosemary St., Chapel Hill, NC 27514

Apple and Apple Pascal are trademarks of the APPLE Computer Corp. IBM and IBM PC are trademarks of International Business Machines. UCSD Pascal is a trademark of the Regents of the University of California. Osborne is a trademark of Osborne Computer. EPSON is a trademark of EPSON America, Inc. C. Itoh is a trademark of C. Itoh Electronics.

Circle no. 76 on reader service card.

Circles (Listing continued, text begins on page 18)

Listing Two

086C 6A	56	ROR	
086D 8D 30 09	57	STA R2	
0870	58 ;		
0870 AD 3C 09	59	LDA YMID	;YCRD<-[YMID-R2]
0873 AA	60	TAX	;X<-YMID FOR LATER
0874 38	61	SEC	
0875 ED 30 09	62	SBC R2	
0878 8D 3D 09	63	STA YCRD	
087B	64 ;		
087B AD 37 09	65	LDA XMIDL	;X2<<-[2*XMID]
087E 0A	66	ASL	
087F 8D 33 09	67	STA X2L	
0882 AD 36 09	68	LDA XMIDH	
0885 2A	69	ROL	
0886 8D 32 09	70	STA X2H	
0889	71 ;		
0889 8A	72	TXA	;A<-YMID
088A 0A	73	ASL	;Y2<<-[2*YMID]
088B 8D 3B 09	74	STA Y2L	
088E A9 00	75	LDA #0	
0890 8D 3E 09	76	STA YCRDF	;YCRDF<-0
0893 2A	77	ROL	
0894 8D 3A 09	78	STA Y2H	
0897	79 ;		
0897 8A	80	TXA	;A<-YMID
0898 18	81	CLC	;XY<<-[XMID+YMID]
0899 6D 37 09	82	ADC XMIDL	
089C 8D 39 09	83	STA XYL	
089F A9 00	84	LDA #0	
08A1 6D 36 09	85	ADC XMIDH	
08A4 8D 38 09	86	STA XYH	
08A7	87 ;		
08A7 AD 37 09	88	LDA XMIDL	;XCRD<<-[XMID-R2]
08AA 38	89	SEC	
08AB ED 30 09	90	SBC R2	
08AE 8D 35 09	91	STA XCRDL	
08B1 AD 36 09	92	LDA XMIDH	
08B4 E9 00	93	SBC #0	
08B6 8D 34 09	94	STA XCRDH	
08B9	95 ;		
08B9 AD 35 09	96	NXPOINT LDA XCRDL	;H<<-XCRD
08BC 8D 2E 09	97	STA HL	
08BF AD 34 09	98	LDA XCRDH	
08C2 8D 2D 09	99	STA HH	
08C5	100 ;		
08C5 AD 3D 09	101	LDA YCRD	;V<<-YCRD
08C8 8D 31 09	102	STA V	
08CB	103 ;		
08CB 20 2C 08	104	JSR PLOTFOUR	;PLOT SET OF POINTS
08CE	105 ;		
08CE AD 39 09	106	LDA XYL	;H<<-[XY-YCRD]
08D1 38	107	SEC	
08D2 ED 3D 09	108	SBC YCRD	

(Continued on page 26)

Z80 Software

SOFTWARE DESCRIPTIONS

TPM (TPM I) - \$80 A Z80 only operating system which is capable of running CP/M programs. Includes many features not found in CP/M such as independent disk directory partitioning for up to 255 user partitions, space, time and version commands, date and time, create FCB, chain program, direct disk I/O, abbreviated commands and more! Available for North Star (either single or double density), TRS-80 Model I (offset 4200H) or II, Versafloppy I, or Tarbell I.

TPM-II - \$125 An expanded version of TPM which is fully CP/M 2.2 compatible but still retains the extra features our customers have come to depend on. This version is super FAST. Extended density capability allows over 600K per side on an 8" disk. Available preconfigured for Versafloppy II (8" or 5"), Epson QX-10, Osborne II or TRS-80 Model II.

CONFIGURATOR I

This package provides all the necessary programs for customizing TPM for a floppy controller which we do not support. We suggest ordering this on single density (8SD).

Includes: TPM-II (\$125), Sample BIOS (BIOS) SOURCE (\$FREE), MACRO II (\$100), LINKER (\$80), DEBUG I (\$80), QED (\$150), ZEDIT (\$50), TOP I (\$80), BASIC I (\$50) and BASIC II (\$100)

\$815 Value

NOW \$250

CONFIGURATOR II

Includes: TPM-II (\$125), Sample BIOS (BIOS) SOURCE (\$FREE), MACRO II (\$100), MACRO III (\$150), LINKER (\$80), DEBUG I (\$80), DEBUG II (\$100), QSAL (\$200), QED (\$150), ZTEL (\$80), TOP II (\$100), BUSINESS BASIC (\$200) and MODEM SOURCE (\$40) and DISASSEMBLER (\$80)

\$1485 Value

NOW \$400

MODEL I PROGRAMMER

This package is only for the TRS-80 Model I. Note: These are the ONLY CDL programs available for the Model I. It includes: TPM I (\$80), BUSINESS BASIC (\$200), MACRO I (\$80), DEBUG I (\$80), ZDDT (\$40), ZTEL (\$80), TOP I (\$80) and MODEM (\$40)

\$680 Value

NOW \$175

MODEL II PROGRAMMER

This package is only for the TRS-80 Model II. It includes: TPM-II (\$125), BUSINESS BASIC (\$200), MACRO II (\$100), MACRO III (\$150), LINKER (\$80), DEBUG I (\$80), DEBUG II (\$100), QED (\$150), ZTEL (\$80), TOP II (\$100), ZDDT (\$40), ZAPPLE SOURCE (\$80), MODEM (\$40), MODEM SOURCE (\$40) and DISASSEMBLER (\$80)

\$1445 Value

NOW \$375

BASIC I - \$50, a 12K+ basic interpreter with 7 digit precision.

BASIC II - \$100, A 12 digit precision version of Basic I.

BUSINESS BASIC - \$200, A full disk extended basic with random or sequential disk file handling and 12 digit precision (even for TRIG functions). Also includes PRIVACY command to protect source code, fixed and variable record lengths, simultaneous access to multiple disk files, global editing, and more!

ACCOUNTING PACKAGE - \$300, Written in Business Basic. Includes General Ledger, Accounts Receivable/Payable, and Payroll. Set up for Hazeltine 1500 terminal. Minor modifications needed for other terminals. Provided in unprotected source form.

MACRO I - \$80, A Z80/8080 assembler which uses CDL/TDL mnemonics. Handles MACROS and generates relocatable code. Includes 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical ops, local and global symbols, linkable module generation, and more!

MACRO II - \$100, An improved version of Macro I with expanded linking capabilities and more listing options. Also internal code has been greatly improved for faster more reliable operation.

MACRO III - \$150, An enhanced version of Macro II. Internal buffers have been increased to achieve a significant improvement in speed of assembly. Additional features include line numbers, cross reference, compressed PRN files, form feeds, page parity, additional pseudo-ops, internal setting of time and date, and expanded assembly-time data entry.

DEVELOPER I

Includes: MACRO I (\$80), DEBUG I (\$80), ZEDIT (\$50), TOP I (\$80), BASIC I (\$50) and BASIC II (\$100)

\$440 Value

NOW \$150

DEVELOPER II

Includes: MACRO II (\$100), MACRO III (\$150), LINKER (\$80), DEBUG I (\$80), DEBUG II (\$100), BUSINESS BASIC (\$200), QED (\$150), TOP II (\$100), ZDDT (\$40), ZAPPLE SOURCE (\$80), MODEM SOURCE (\$40), ZTEL (\$80), and DISASSEMBLER (\$80).

\$1280 Value

NOW \$350

DEVELOPER III

Includes: QSAL (\$200), QED (\$150), BUSINESS BASIC (\$200), ZTEL (\$80) and TOP II (\$100)

\$730 Value

NOW \$300

COMBO

Includes: DEVELOPER II (\$1280), ACCOUNTING PACKAGE (\$300), QSAL (\$200) and 6502X (\$150)

\$1930 Value

NOW \$500

LINKER - \$80, A linking loader for handling the linkable modules created by the above assemblers.

DEBUG I - \$80, A tool for debugging Z80 or 8080 code. Disassembles to CDL/TDL mnemonics compatible with above assemblers. Traces code even through ROM. Commands include Calculate, Display, Examine, Fill, Goto, List, Mode, Open File, Put, Set Wait, Trace, and Search.

DEBUG II - \$100, A superset of Debug I. Adds Instruction Interpreter, Radix change, Set Trap/Conditional display, Trace options, and Zap FCB.

6502X - \$150, A 6502 cross assembler. Runs on the Z80 but assembles 6502 instructions into 6502 object code! Similar features as our Macro assemblers.

QSAL - \$200, A SUPER FAST Z80 assembler. Up to 10 times faster than conventional assemblers. Directly generates code into memory in one pass but also to offset for execution in its own memory space. Pascal like structures: repeat, until, if, then, else, while, do, begin, end, case, of. Multiple statements per line, special register handling expressions, long symbol names, auto and modular assembly, and more! This one uses ZILOG Mnemonics.

QED - \$150, A screen editor which is both FAST and easy to learn. Commands include block delete, copy, and move to a named file or within text, repeat previous command, change, locate, find at start of line, and numerous cursor and window movement functions. Works with any CRT having clear screen, addressable cursor, clear to end of line, clear to end of screen, and 80X24.

DISK FORMATS

When ordering software specify which disk format you would like.

CODE	DESCRIPTION
8SD	8" IBM 3740 Single Density (128 bytes/26 sectors/77 tracks)
8DD	8" Double Density (256 bytes/26 sectors/77 tracks)
8XD	8" CDL Extended Density (1024 bytes/8 sectors/77 tracks = 616K)
5SD	5.25" Single Density (TRS80 Model I, Versafloppy I, Tarbell I)
5EP	5.25" Epson Double Density
5PC	5.25" IBM PC Double Density
5XE	5.25" Xerox 820 Single Density
5OS	5.25" Osborne Single Density
5ZA	5.25" Z80 Apple (Softcard compatible)

TPM INFO

CODE	DESCRIPTION
TPM I:	
NSSD/H	North Star Single Density for Horizon I/O
NSSD/Z	North Star Single Density for Zapple I/O
NSDD/H	North Star Double Density for Horizon I/O
NSDD/Z	North Star Double Density for Zapple I/O
TRS80I	TRS-80 Model I (4200H Offset)
TRS80II	TRS-80 Model II
VII8	Versafloppy I 8"
VII5	Versafloppy I 5.25"
VII8	Versafloppy II 8" (XD)
VII5	Versafloppy II 5.25"
TRS80II	TRS-80 Model II (XD)

Prices and Specifications subject to change without notice.

TPM, Z80, CP/M, TRS80 are trademarks of CDL, Zilog, DRI and Tandy respectively.

ZTEL - \$80, An extensive text editing language and editor modelled after DEC's TECO.

ZEDIT - \$50, A mini text editor. Character/line oriented. Works well with hardcopy terminals and is easy to use. Includes macro command capability.

TOP I - \$80, A Text Output Processor for formatting manuals, documents, etc. Interprets commands which are entered into the text by an editor. Commands include justify, page number, heading, subheading, centering, and more.

TOP II - \$100, A superset of TOP I. Adds: embedded control characters in the file, page at a time printing, selected portion printing, include/merge files, form feed/CRLF option for paging, instant start up, and final page ejection.

ZDDT - \$40, This is the disk version of our famous Zapple monitor. It will also load hex and relocatable files.

ZAPPLE SOURCE - \$80, This is the source to the SMB ROM version of our famous Zapple monitor. It can be used to create your own custom version or as an example of the features of our assemblers. Must be assembled using one of our assemblers.

MODEM - A communication program for file transfer between systems or using a system as a terminal. Based on the user group version but modified to work with our SMB board or TRS-80 Models I or II. You must specify which version you want.

MODEM SOURCE - \$40, For making your own custom version. Requires one of our Macro Assemblers.

DISASSEMBLER - \$80, Does bulk disassembly of object files creating source files which can be assembled by one of our assemblers.

HARDWARE

S-100 — SMB II Bare Board \$50, "System Monitor Board" for S-100 systems. 2 serial ports, 2 parallel ports, cassette interface, 4K memory (ROM, 2708 EPROM, 2114 RAM), and power on jump. When used with Zapple ROM below, it makes putting a S-100 system together a snap.

Zapple ROM \$35, Properly initializes SMB I/II hardware, provides a powerful debug monitor.

IBM PC — Big Blue Z80 board \$595, Add Z80 capability to your IBM Personal Computer. Runs CP/M programs but does not require CP/M or TPM. Complete with Z80 CPU, 64K add on memory, serial port, parallel port, time and date clock with battery backup, hard disk interface, and software to attach to PC DOS and transfer programs. Mfr'd by QCS.

50% Discount on all CDL software ordered at the same time as a Big Blue (and for the Big Blue).

APPLE II — Chairman Z80 \$345, Add Z80 capability to your Apple II/II Plus computer. Runs CP/M programs with our more powerful TPM. Includes 64K memory add on (unlike the competition this is also useable by the 6502/DOS as well as the Z80), TPM, QSAL assembler, QED Screen Editor, and Business Basic. Mfr'd by AMT Research.

Apple Special \$175, Buy the Apple Z80 Developer at the same time as the "Chairman" and pay only \$175 instead of \$325.

APPLE Z80 DEVELOPER

Includes: 6502X (\$150), MACRO II (\$100), MACRO III (\$150), QSAL (\$200), QED (\$150), LINKER (\$80), DEBUG I (\$80), DEBUG II (\$100), ZDDT (\$40) and BUSINESS BASIC (\$200)

VALUE: \$1250

NOW \$325

\$175 when purchased with AMT "Chairman" Board

ORDERING INFORMATION:

VISA/MasterCard/C.O.D.

Call or Write With Ordering

Information....



OEMS:

Many CDL products are available for licensing to OEM's. Write to Carl Galletti with your requirements.

Dealer Inquiries Invited.

For Phone Orders ONLY Call Toll Free...

1-(800) 458-3491

(Except Pa.)

Ask For Extension #15

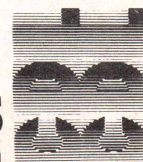
For information and Tech Queries call

(609) 599-2146

Computer Design Labs

342 Columbus Avenue/Trenton, NJ 08629

Circle no. 21 on reader service card.



Circles (Listing continued, text begins on page 18)

Listing Two

08D5 8D 2E 09	109		STA HL	
08D8 AD 38 09	110		LDA XYH	
08DB E9 00	111		SBC #0	
08DD 8D 2D 09	112		STA HH	
08E0	113	;		
08E0 AD 39 09	114		LDA XYL	;V<-[XY-XCRD]
08E3 38	115		SEC	
08E4 ED 35 09	116		SBC XCRDL	
08E7 8D 31 09	117		STA V	
08EA	118	;		
08EA 20 2C 08	119		JSR PLOTFOUR	;PLOT REMAINING POINTS
08ED	120	;		
08ED AD 3C 09	121		LDA YMID	;DIVISOR<-[YMID-YCRD]
08F0 38	122		SEC	
08F1 ED 3D 09	123		SBC YCRD	
08F4 85 1A	124		STA DIVISOR	
08F6	125	;		
08F6 AD 37 09	126		LDA XMIDL	;DIVIDEND<-[XMID-XCRD]*256
08F9 38	127		SEC	
08FA ED 35 09	128		SBC XCRDL	
08FD	129	;		
08FD A2 08	130		LDX #8	;BITCT<-8
08FF	131	;		
08FF A0 00	132		LDY #0	;CLEAR QUOTIENT
0901 84 1B	133		STY QUOTIENT	
0903	134	;		
0903 06 1B	135	DIVIDE1	ASL QUOTIENT	
0905 2A	136		ROL	
0906 C5 1A	137		CMP DIVISOR	
0908 90 04	138		BCC DIVIDE2	
090A E5 1A	139		SBC DIVISOR	
090C E6 1B	140		INC QUOTIENT	
090E CA	141	DIVIDE2	DEX	
090F D0 F2	142		BNE DIVIDE1	
0911	143	;		
0911 AD 3E 09	144		LDA YCRDF	;YCRDF<-[YCRDF-QUOTIENT]
0914 38	145		SEC	
0915 E5 1B	146		SBC QUOTIENT	
0917 8D 3E 09	147		STA YCRDF	
091A	148	;		
091A B0 03	149		BCS CKX	;YCRDY<-1?
091C CE 3D 09	150		DEC YCRD	;YES,YCRD<-[YCRD-1]
091F	151	;		
091F EE 35 09	152	CKX	INC XCRDL	;XCRD<-[XCRD+1]
0922 D0 03	153		BNE CKX1	
0924 EE 34 09	154		INC XCRDH	
0927	155	;		
0927 CE 30 09	156	CKX1	DEC R2	;TALLY R2
092A 10 8D	157		BPL NXPOINT	;REPEAT UNTIL XCRD=XMID
092C 60	158		RTS	
092D	159	;		
092E	160	HH	DFS 1	;HORIZONTAL PLOT VALUE
092F	161	HL	DFS 1	

(Continued on page 28)

USE MAGIC/L™ TO WRITE JUST ONE PROGRAM



Bet you can't stop with one.

For just \$295 MAGIC/L™ is yours. And suddenly you find you've bought more than a language. MAGIC/L is an entire interactive environment; an assembler, a compiler, an editor, an I/O package, and a system call facility are all wrapped up and delivered. And now MAGIC/L is available for CP/M-based computers.

MAGIC/L is easy to learn. It has syntax similar to C and Pascal, and because it's extensible as well as interactive, it dramatically increases productivity.

Program development features include a built-in text editor, command line recall, CCP, STAT, and PIP command emulation, and the ability to store keyboard dialog on disks.

Key language features include: CHAR, INTEGER, LONG, REAL, and String data types; record structures similar to the STRUCT facility in C; and a complete I/O package that can provide random access, variable length I/O to any CP/M file.

And MAGIC/L offers great portability. Source code which runs on CP/M can be compiled unmodified and run on any other processor.

Typical applications include hardware interfacing, process control, games creation, interactive graphics and image processing. MAGIC/L has made programming easier for DEC, 68000, and Data General users. Now it's working for CP/M users too.

MAGIC/L provides everything you need to write a complete program. But the only way to be convinced is to try it yourself. Send us your \$295 check or money order—we also accept MasterCard and VISA—we'll send MAGIC/L for CP/M to you at once. A full money back guarantee is part of the package. Once you've sampled that first program, you'll have to try another . . . and another . . . and another. MAGIC/L . . . it's more than a language.

MAGIC/L . . . It's more than a language



LOKI ENGINEERING, INC.

55 Wheeler St., Cambridge, MA 02138 (617) 576-0666

Circles

(Listing continued, text begins on page 18)

Listing Two

0930	162	R	DFS 1	; RADIUS
0931	163	R2	DFS 1	; HOLDS R/SQR(2)
0932	164	V	DFS 1	; VERTICAL PLOT
0933	165	X2H	DFS 1	; HOLDS XMID*2
0934	166	X2L	DFS 1	
0935	167	XCRDH	DFS 1	; X COORDINATE
0936	168	XCRDL	DFS 1	
0937	169	XMIDH	DFS 1	; HORIZONTAL CENTER
0938	170	XMIDL	DFS 1	
0939	171	XYH	DFS 1	; HOLDS XMID+YMID
093A	172	XYL	DFS 1	
093B	173	Y2H	DFS 1	; HOLDS YMID*2
093C	174	Y2L	DFS 1	
093D	175	YMID	DFS 1	; VERTICAL CENTER
093E	176	YCRD	DFS 1	; Y COORDINATE
093F	177	YCRDF	DFS 1	; FRACTIONAL PART OF YCRD
093F	178	;		
001A	179	DIVISOR	EPZ \$1A	
C052	180	FULL	EQU \$C052	
C050	181	GRAPHICS	EQU \$C050	
00E4	182	HCOLOR	EPZ \$E4	
C057	183	HIRES	EQU \$C057	
F457	184	HPLLOT	EQU \$F457	; APPLESOFT HIRES PLOT
00E6	185	HPAGE	EPZ \$E6	
00E1	186	LASTHH	EPZ \$E1	; HORIZ COORD OF LAST HPLLOT
00E0	187	LASTHL	EPZ \$E0	
00E2	188	LASTV	EPZ \$E2	; VERT COORD OF LAST HPLLOT
C054	189	PAGE1	EQU \$C054	
001B	190	QUOTIENT	EPZ \$1B	
093F	191	;		
093F	192		END	

***** END OF ASSEMBLY

End Listing Two

(Listing Three begins on page 30)

Programmers... Now you can "C" into your programs!

C Compiler

Runs on the IBM® Personal Computer or MSDOS® compatible computers. Provides features and flexibility needed for both systems and application programming. Makes it possible to test programs at the C source level. The software program features full documentation, generates fast code, and has a standard C function library.

- All C language elements for operations on characters, signed and unsigned integers, structures, and unions
- Outputs IBM PC assembler source code
- PL/M compatibility mode
- Embedded shell in compiled output supports argc, argv, and I/O redirection
- Complete preprocessor
- Allows more than 64K of code; extended pointers allow access to more than 64K of data

c-window™ Source Level Debugger

Designed for sophisticated programmers who want more capability than inserting "printf" statements or using assembler level debugging... new programmers who want to better understand how C operates... trainers who teach the C language. It helps you find the precise statement in a program where an error occurs and observe when data values are changed in a program.

- True source level single step
- Auto and external variable display and modification in a variety of formats
- Full C expression evaluation and recursion
- Varied breakpoints
 - at function name and line number
 - sticky and non-sticky
 - at function whenever an expression evaluates as TRUE
- Multiple variable display as program is tested

c-systems

The following is an excerpt from a typical c-window debugging session. The ">>>" is the c-window prompt, and the underscored text is the operator input. The text on the right is a brief description of the operations taking place.

entry at main line 15

>>>bs addline, 58

>>>g

break at addline line 58

>>>d ptr

bae

>>>d ptr = root

f02

>>>d ++ptr

f14

>>>ds ptr->symname

xyzzz

>>>cs 1 d i

>>>s

step at addline line 59

<1> d i

11

>>>d i==j

1

>>>bx addline,2,i<j&& k==0.

- c-window entry prompt showing function name and line number
- A breakpoint is set in function addline, line 58.
- The "go" command starts execution.
- The breakpoint is reached.
- The value of variable "ptr" is displayed.
- The variable "ptr" is set to the value of the variable "root".
- The variable "ptr" is incremented to the next entry in the table.
- Display the string at the member "symname" in the structure pointed to by "ptr".
- Command set. The variable "i" will be displayed for each break in execution.
- Single step.
- c-window displays the next line to be executed.
- c-window executes the automatic command, showing the value of "i" to be 11 hex.
- Test if "i" has the same value as "j".
- They are equal.
- Set an expression break. Once execution begins, c-window will break execution on the second occurrence of the expression "i<j&&k==0" evaluating non-zero in the function "addline".

For the finest time-saving software contact:

c-systems

We make
programming
easier

P.O. Box 3253 • Fullerton, CA 92634 • (714) 637-5362

Name _____

Address _____

I would like to order:

- ☐ C Compiler..... \$195.00
- ☐ c-window (requires C Compiler)..... \$195.00
- ☐ Demo package (includes diskette, manual)
price applicable toward purchase of above... \$45.00

Require 128K and two disk drives.
c-window is a trademark of C-Systems. IBM is a
trademark of IBM Corp. and MSDOS is a trademark
of MicroSoft.
Prices subject to change without notice.

Circles (Listing continued, text begins on page 18)

Listing Three

```
10  ' *** CIRCLE PLOT ***
20  '
30  INPUT "CENTER, RADIUS"; CX,CY,R: X=R: Y=0: A=-2*X+1:
    B=1: GOSUB 70: GOTO 30
40  '
50  ' PLOT A POINT IN EACH OCTANT
60  '
70  PSET(X+CX,Y+CY): PSET(Y+CX,X+CY): PSET(-Y+CX,X+CY):
    PSET(-X+CX,Y+CY): PSET(-X+CX,-Y+CY): PSET(-Y+CX,-X+CY):
    PSET(Y+CX,-X+CY): PSET(X+CX,-Y+CY)
80  '
90  ' COMPUTE NEXT POINT.  F IS  $X^2+Y^2-R^2$ , A IS THE CHANGE
100 ' IN  $X^2$  WHEN X IS DECREMENTED BY 1, AND B IS THE CHANGE
110 ' IN  $Y^2$  WHEN Y IS INCREMENTED BY 1.  F IS NOT ALLOWED TO
120 ' EXCEED R; EQUIVALENTLY, THE POINT (X,Y) IS KEPT WITHIN
130 ' A DISTANCE  $R+1/2$  OF THE CIRCLE CENTER.  THE ALGORITHM
140 ' IS DONE WHEN THE CHANGE IN  $Y^2$  REACHES THE NEGATIVE OF
150 ' THE CHANGE IN  $X^2$  (  $B \geq -A$  ).
160 '
170 IF  $B \geq -A$  THEN RETURN ELSE  $Y=Y+1$ :  $F=F+B$ : IF  $F > R$  THEN
     $F=F+A$ :  $A=A+2$ :  $X=X+1$ 
180  $B=B+2$ : GOTO 70
```

End Listing Three

Circles

Listing Four

```
10 DEFINT A-Z
20 PRINT CHR$(26) 'CLEAR DUMB TTY SCREEN
50 FOR I=1 TO 11
55 AE=I*2 'WIDTH OF ELLIPSE
56 BE=I*1 'HEIGHT OF ELLIPSE
57 XC=I*4+1 'CENTER.X OF ELLIPSE
58 YC=I*1 'CENTER.Y OF ELLIPSE
60 GOSUB 1060 'PLOT A CIRCLE
70 NEXT I 'PLOT 11 CIRCLES
998 END
1050 '***** CIRCLE SUBROUTINE
1060 XF=0 'INIT X-OFFSET
1070 YF=BE 'INIT Y-OFFSET
1080 XD=BE*BE 'INIT COMPUTATION OF X-SQUARED
1090 YD=(2*BE-1)*AE*AE 'INIT COMPUTATION OF Y-SQUARED
1100 DX=2*BE*BE 'DEFINE DELTA-(X-SQUARED)
1110 DY=2*AE*AE 'DEFINE DELTA-(Y-SQUARED)
1120 ER=0 'INIT ERROR (I.E.  $ER=AE^2*BE^2-XF^2*BE^2-YF^2*AE^2$ )
1130 GOSUB 1260 'PLOT THE FOUR POINTS
1140 TX=ER+XD
```



```

      : TY=ER-YD
      : TB=ER+XD-YD
1150 IF ABS(TX) >=ABS(TY) OR ABS(TX) >=ABS(TB) THEN 1170
1160 XF=XF+1
      : ER=TX
      : XD=XD+DX
      : GOTO 1220
1170 IF ABS(TY) >=ABS(TX) OR ABS(TY) >=ABS(TB) THEN 1190
1180 YF=YF-1
      : ER=TY
      : YD=YD-DY
      : GOTO 1220
1190 IF ABS(TB) >=ABS(TX) OR ABS(TB) >=ABS(TY) THEN 1210
1200 XF=XF+1
      : YF=YF-1
      : ER=TB
      : YD=YD-DY
      : XD=XD+DX
      : GOTO 1220
1210 PRINT"OOPS"; 'IF HERE THEN THERE IS A BUG.
1220 GOSUB 1260 'PLOT THE POINTS
1230 IF YF<>0 THEN 1140
1240 RETURN
1250 '***** ROUTINE TO PLOT FOUR POINTS AT ONCE
1260 XP=XC+XF
      : YP=YC+YF
      : GOSUB 1320
1270 XP=XC+XF
      : YP=YC-YF
      : GOSUB 1320
1280 XP=XC-XF
      : YP=YC+YF
      : GOSUB 1320
1290 XP=XC-XF
      : YP=YC-YF
      : GOSUB 1320
1300 RETURN
1310 '***** ROUTINE TO PLOT A POINT ON A DUMB TERMINAL
1320 C1=YP+32
      : C2=XP+32
1330 IF YP<0 OR YP>23 OR XP<0 OR XP>79 THEN 1360
1340 PRINT CHR$(27);CHR$(61);CHR$(C1);CHR$(C2);"*";
1350 RETURN
1360 PRINT "POINT OUT OF BOUNDS"
      : STOP

```

End Listing Four

How FORTH level 2 makes your PC 412% more powerful

What used to take 40 hours can now be done in 7.8 hours — that's 412% — and it is only one way of measuring power. There are others — convenience, debugging ability, run-time, compile-time, memory size . . . name it! FORTH level 2 will increase your power because it provides an interactive, integrated environment which promotes immediate debugging and modular design. Other benefits such as real-time performance depend on the quality of the implementation. Not all FORTHs are equal, so compare before buying. This one is a subset of polyFORTH, designed by FORTH, INC., tested and proven in the field for more than 5 years.

At last it is available at an affordable price of **\$149** — and features the following:

■ Multitasking

Any number of background tasks. Concurrent operation of monochrome and color monitor. Concurrent printer operation.

■ Compatibility

IBM DOS file interface utility allows access to files created under IBM DOS with FORTH's improved performance and power.

■ On-Line Documentation

As well as over 700 pages of supporting documentation including *Starting FORTH* by Leo Brodie and 360-page *User's Manual*.

■ Turnkey-compiler™

Utility for making binary turnkey applications. Such programs can be resold without license fee under specific conditions.

Options:

■ 8087 support	\$129
■ Data Base support	\$75
■ Report Writer	\$25
■ Documentor	\$75
■ Multi-User capability	\$129
■ Graphics module	\$100

This is not a complete list and you may have questions not covered in this material — give us a call! We will address them and send you a brochure.

Act now! After all, if this is the right software tool for you, shouldn't you know about it?

Forth Technology

432 15th Street ■ Santa Monica, CA 90402

(213) 372-8493

Circle no. 40 on reader service card.

Cursor Control for Dumb Terminals

A friend of mine asked me one day, "Why should I purchase a new CRT terminal when I can buy a surplus unit for less than half the price?"

"Because," I answered, "that so-called bargain of yours is what we used to call a 'dumb' terminal. Unlike today's marvels, it has only the most basic of cursor control functions: home, line feed, carriage return, backspace, and skip (the skip function moves the cursor across the screen one column at a time without deleting any currently at that position). It is incapable of direct cursor addressing, which is required by most of the software available on the market today. In short, it is not a bargain; it is an antique."

"What," my friend asked, "should I do with this antique I just bought?"

The solution to his problem turned out to be surprisingly simple and should be of interest to anyone owning a terminal without direct cursor addressing and a microcomputer system running CP/M. (It should also interest those wanting to connect more esoteric peripherals than terminals or printers to their systems, if only to illustrate the level of complexity that can be supported.) Only the software need be modified. Even then the CP/M operating system remains absolutely standard and is still capable (assuming the IOBYTE feature has been implemented) of supporting a second terminal with direct cursor addressing without further modifications. What more could you ask for?

A terminal with direct cursor addressing traps a predefined string of ASCII characters sent from the computer to the terminal and uses the information contained in this string to position the cursor on the screen. (The string itself, of course, is not displayed.) As an example, the Televideo 950 terminal responds to the ASCII string "ESC=yx," where y is an ASCII character from "SP" (32 decimal) to "7" (55 decimal) that positions the cursor vertically and x is an ASCII character from "SP" to "o" (111 decimal) that positions the cursor horizontally. To emulate this feature, a "dumb" terminal should be able to respond to a similar

string of ASCII characters and position its cursor accordingly.

While it would be nice to have the terminal perform this function through its own hardware, this is obviously impractical for a large number of users; re-designing and rewiring old printed circuit boards without schematics is a task few care to tackle! What remains is to modify the software, specifically CP/M's BIOS (Basic Input/Output System), which provides the interface between the terminal and the host computer. The CP/M operating system can then monitor what the running program is sending through the BIOS to the terminal.

When an ASCII "ESC" character is sent to the modified BIOS (from, say, a full-screen word processing program), it will store this character temporarily rather than output it through the serial port to the terminal. When the next character sent by the program is received, the BIOS will check to see if it is an ASCII "=" character. If it isn't, the BIOS will output both the stored "ESC" character and the received character to the terminal, then continue to output characters received from the program until the next "ESC" character arrives; otherwise, it will recognize the two characters as the start of a direct cursor positioning command, discard them, and wait for the next character.

Once this character is received, the BIOS will decode it, output a single ASCII control character to home the cursor on the terminal, and output the required number of line feeds to position the cursor vertically. Then it will trap the next character sent by the program, decode it, and output the required number of skip control characters to position the cursor horizontally. Once the cursor is positioned where required, everything returns to normal: all ASCII characters sent to the BIOS by the program will be sent to the terminal after being monitored for an "ESC" character and the possible start of another direct cursor positioning command.

It sounds complicated, but it is simple to program and works beautifully as long as you operate the terminal at 9600 baud. At this speed the cursor will take, in the worst case, approximately one-tenth of a second to traverse the screen (from the home position to the 24th row and 80th column). The motion of the cursor is perceptible but not objectionable; a slower baud rate, however, is an invitation to crossed eyeballs from watch-

ing the cursor bounce around the screen!

The entire software routine can be written as a peripheral driver and added to your BIOS in exactly the same way as you would add a printer or modem to your system.

If your version of CP/M doesn't include the IOBYTE feature, then the accompanying code (Listing One, page 33) should be substituted for your existing terminal driver (usually the physical device TTY: or CRT: in the BIOS source code). Once this is done, your version of CP/M will be customized for the "dumb" terminal being used as its console device. When a newer model terminal is purchased, the BIOS must be customized once again to support it as the console device instead of the old terminal.

Fortunately, most commercial versions of CP/M do include the IOBYTE feature. In this case, the accompanying code should be placed in whatever physical I/O device driver routine isn't being used (usually the user-defined console device UC1:). By leaving the TTY: or CRT: physical device code (whichever currently supports the computer's terminal device) intact or by modifying it to support the terminal you eventually intend to purchase, you can switch between a "dumb" terminal and a modern direct-cursor-addressable terminal simply by using the `STAT CON:=UC1:` and `STAT CON:=TTY:` (or `STAT CON:=CRT:`, as applicable) commands. You can also modify your BIOS to initially set the IOBYTE so that `CON:=UC1:` after each cold boot.

Because the procedure for modifying CP/M's BIOS may vary, depending on the idiosyncracies of your particular system, I will not describe it in this article. If you haven't interfaced a peripheral to your system before, I suggest that you have your vendor or an experienced friend perform the modifications indicated in Listing One; otherwise, you should consult some of the excellent books on CP/M that are available (the finest I know of is David Cortesi's *Inside CP/M - A Guide for Users and Programmers*, CBS College Publishing, 1982).

By the way, some excellent bargains are available on the surplus computer equipment market - if you know exactly what you are buying. Otherwise, caveat emptor, or "let the buyer beware!" ■■■

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 197.

by Ian Ashdown

Ian Ashdown, byHeart Software, 2-2016
West First Avenue, Vancouver, B.C., V6J
1G8, Canada.

Cursor Control Listing (Text begins on page 32)

```
; *****
;
;                               CP/M BIOS Modifications
;
;   Author: Ian Ashdown
;           byHeart Software
;           2 - 2016 West First Avenue
;           Vancouver, B.C. V6J 1G8
;
;   Date:   February 10th, 1983
;
; *****
;
; The following code has been added to allow the ANAND II "dumb"
; terminal to emulate a direct-addressable cursor terminal
; (specifically the TELEVIDEO 950). It does this by trapping an
; "ESC=yx" control code from a program and interpreting "y" as
; the vertical row address (where 20H is row #1) and "x" as the
; horizontal column address of the cursor (where 20H is column
; #1). The BIOS then homes the cursor and outputs as many line
; feeds and skips as is required to position the cursor without
; changing the current screen display. At 9600 baud the worst
; case transit time of the cursor (from "home" to the 24th row
; and 80th column) is approximately 1/10 second.
;
; ***** EQUATES *****
;
; NOTE: The following equates are independent of the system used.
;       However, they may already be defined in your BIOS source
;       code.
;
LF      EQU      0AH      ;ASCII line feed
CR      EQU      0DH      ;ASCII carriage return
ESC     EQU      1BH      ;ASCII escape
;
; NOTE: The following equates are specific to my California
;       Computer System's serial I/O port and the control codes
;       recognized by the ANAND II terminal. Replace them
;       with those required by your system.
;
RESET   EQU      19H      ;ANAND II "home cursor" control char.
SKIP    EQU      14H      ;ANAND II "cursor forward" control char.
SDATA   EQU      20H      ;SERIAL DATA PORT
SLSTAT  EQU      SDATA+5  ;Serial line status port
RXRDY   EQU      0000001B ;Receive data available bit
TXMTY   EQU      0010000B ;Transmit buffer empty bit
;
; ***** NEW PHYSICAL DEVICE UC1: DRIVER ROUTINE *****
;
; I/O DRIVERS FOR THE 8250 ASYNC COMM ELEMENT
; (PHYSICAL DEVICE UC1:)
;
```

(Continued on next page)

Cursor Control Listing (Listing continued, text begins on page 32)

```
; NOTE: The following CUST1: and CUSI1: subroutines are vendor-
; dependent code (see COMMENTARY at the end of this
; listing), and should be replaced with the TTY: or CRT:
; driver code specific to your system.
;
CUST1: IN      SLSTAT  ;Get 8250 line status
      ANI      RXRDY   ;See if receive data available
      RZ                ;Return if not
      ADI      OFFH AND NOT RXRDY      ;Flag that data is
      RET                                ;available
;
CUSI1: CALL    CUST1   ;Get 8250 line status
      JZ      CUSI1   ;Loop until data is in
      IN      SDATA   ;Read the data
      RET
;
; NOTE: The following subroutines CUSO1: through HORZOUT:
; inclusive are independent of the system used.
;
CUSO1: LDA     ESCPTR  ;Get escape sequence pointer
      CPI     0        ;Jump to TESTCH if this isn't an
      JZ      TESTCH   ;escape sequence
      CPI     1        ;Jump to TESTEQU if this is an escape
      JZ      TESTEQU  ;sequence
      CPI     2        ;Jump to VERTIN if true (character is
      JZ      VERTIN   ;row address of cursor)
      JMP     HORZIN   ;Jump to HORZIN (character is column
                        ;address of cursor)
TESTCH: MOV     A,C      ;Get character from register C
      CPI     ESC       ;Jump to NOTESC if it isn't an ASCII ESC
      JNZ     NOTESC
      MVI     A,1       ;Set escape sequence pointer to 1 to
      STA     ESCPTR    ;indicate ASCII ESC character
      RET
NOTESC: CALL    CUSOUT  ;Output character
      RET
;
TESTEQU: MOV    A,C      ;Get character from register C
      CPI     '='       ;Jump to NOTEQU if it isn't an ASCII "="
      JNZ     NOTEQU
      MVI     A,2       ;Set escape sequence pointer to 2 to
      STA     ESCPTR    ;indicate ASCII "=" character
      RET
NOTEQU: MVI     A,0      ;Set escape sequence pointer to 0 (false
      STA     ESCPTR    ;alarm)
      MVI     A,ESC     ;Output previous ASCII ESC character
      CALL    CUSOUT
      MOV     A,C      ;Get current character from register C
      CALL    CUSOUT   ;Output it
      RET
;
```

(Continued on page 36)

Compilers and **C**ross compilers

TELECON'S C COMPILERS OFFER YOU

- FULL C
- UNIX* Ver. 7 COMPATABILITY
- NO ROYALTIES ON GENERATED CODE
- GENERATED CODE IS REENTRANT
- C AND ASSEMBLY SOURCE MAY BE INTERMIXED
- UPGRADES & SUPPORT FOR 1 YEAR

IN THESE CONFIGURATIONS:

HOST	6809 TARGET	PDP-11*/LSI-11* TARGET	8080/(Z80) TARGET	8088/8086 TARGET
FLEX*/UNIFLEX* OS-9*	\$200.00 WITHOUT FLOAT \$350.00 WITH FLOAT	500.00	500.00	500.00
RT-11*/RSX-11* PDP-11*	500.00	200.00 WITHOUT FLOAT 350.00 WITH FLOAT	500.00	500.00
CP/M* 8080/(Z80)	500.00	500.00	200.00 WITHOUT FLOAT 350.00 WITH FLOAT	500.00
PCDOS*/CP/M86* 8088/8086	500.00	500.00	500.00	200.00 WITHOUT FLOAT 350.00 WITH FLOAT

Others Pending

C SOURCE AVAILABLE FOR \$2,500⁰⁰

SO ... IF YOU'RE READY TO MOVE UP TO C ...

CALL

408-275-1659

TELECON SYSTEMS

1155 Meridian Avenue, Suite 218
San Jose, California 95125

*PCDOS is a trademark of IBM CORP. MSDOS is a trademark of MICROSOFT. UNIX is a trademark of BELL LABS. RT-11/RSX-11/PDP-11 is a trademark of Digital Equipment Corporation. FLEX/UNIFLEX is a trademark of Technical Systems consultants. CP/M & CP/M86 are trademarks of Digital Research. OS-9 is a trademark of Microware & Motorola.

Circle no. 106 on reader service card.

Cursor Control Listing (Listing continued, text begins on page 32)

```

VERTIN: MOV    A,C      ;Get character from register C
        SUI    1FH      ;Subtract cursor address bias (20H is row
                        ;number 1) to get row address counter
        MOV    C,A      ;Store row address counter in register C
        MVI    A,3      ;Set escape sequence pointer to 3
        STA    ESCPTR
        MVI    A,RESET  ;Output "reset cursor" control character
        CALL   CUSOUT
VERTOUT: DCR    C        ;Decrement row address
        RZ      ;Return if done
        MVI    A,LF      ;Output a line feed
        CALL   CUSOUT
        JMP    VERTOUT  ;Loop until done

;
HORZIN: MOV    A,C      ;Get character from register C
        SUI    1FH      ;Subtract cursor address bias (20H is
                        ;column number 1) to get column counter
        MOV    C,A
        XRA    A        ;Set escape sequence pointer to 0
        STA    ESCPTR
HORZOUT: DCR    C        ;Decrement column counter
        RZ      ;Return if done
        MVI    A,SKIP   ;Output a "cursor skip" control character
    
```

Huge Discounts

on Systems, Components and Peripherals

WELCOME TO ONE STOP SHOPPING: We offer a wide range of systems, peripherals and software at discounts of up to 34% off list price. Send \$4 for catalogue replete with technical specifications, and other interesting information and receive a \$10 discount coupon for your first purchase. A sampling of our discounts includes: GODBOUT 816 A \$4,030

U.S. ROBOTICS PASSWORD: \$368

HAZELTINE ESPRIT I: \$480

IMS 8000: \$3,356

NEC APC H02: \$2,758

SYSTEMS: CompuPro/Godbout, EPSON QX10, IMS International, Intercontinental Micro, Lomas Data Products, NEC APC, Seattle Computer, Systems Group, Tarbell Electronics.

PRINTERS: Epson, Mannesman Tally, NEC Spinwriters, Texas Instruments, Teletype Corp.

GRAPHICS: Autocad Software, Houston Instruments Digitizer and Plotters, Scion MicroAngelo.

PLUS: CORVUS AND QCS WINCHESTERS, U.S. ROBOTICS MODEMS, SOFTWARE, 3M DISKETTES, IBM 3270 NETWORK TERMINALS AND PRINTERS, QUADBOARD.

*** WE EXPORT TO ALL COUNTRIES ***

JOHN D. OWENS ASSOCIATES, INC.

DEPT. DD 12 SCHUBERT STREET
STATEN ISLAND, NEW YORK 10305

(212) 448 6283 (212) 448 6298 (212) 448 2913

TWX: 710 588 2844 CABLE: OWENSASSOC.

Prices subject to change without notice. We have no reader inquiry number.

Circle no. 75 on reader service card.



LEO ELECTRONICS, INC.

8921 S. Sepulveda # 208

Los Angeles, CA. 90045

(213) 641-3101 (800) 421-2418

TLX: 664-688 Interline LSA

PRICE !

QUALITY !

SERVICE !

RAMS

4116 (150ns)	1.35)	16K UPGRADE
4116 (200ns)	1.25)	
4164 (150ns)	4.95)	64K UPGRADE
4164 (200ns)	4.75)	
6116P-3	4.40	

EPROMS

2708	3.00	2532	4.50
2716	3.20	2732	3.95
TMS2716	4.75	2764	7.00

TERMS: Check, Visa, Mastercard. Call for C.O.D.
U.S. Funds only. California residents add 6½% sales tax.

SHIPPING: Add \$2.00 for Ground and \$5.00 for Air.

**ALL MAJOR MANUFACTURERS
ALL PARTS 100% GUARANTEED**

Pricing subject to change without notice.

Circle no. 58 on reader service card.


```

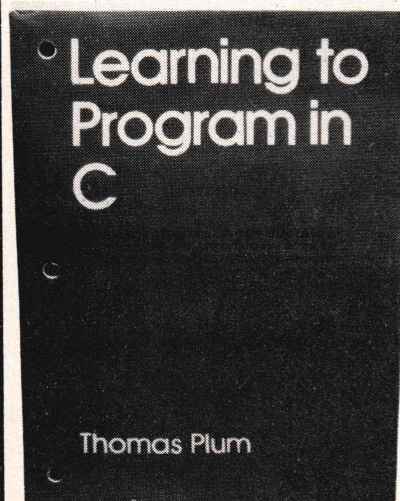
CALL    CUSOUT
JMP     HORZOUT ;Loop until done
;
; ESCPTR DB      0      ;Escape sequence pointer
;
; NOTE: The following CUSOUT: subroutine is vendor-dependent
;       code (see COMMENTARY at the end of this listing), and
;       should be replaced with the TTY: or CRT: driver code
;       specific to your system.
;
CUSOUT: PUSH    PSW      ;Save character
CUSOUT1: CALL   CUSOST   ;Get 8250 line status
        JZ      CUSOUT1 ;Wait until one of the registers empties
        POP     PSW      ;Recall character
        OUT     SDATA    ;Output the data
        RET
;
CUSOST: IN      SLSTAT   ;Get 8250 line status
        ANI     TXMTY    ;Isolate TX buffer empty bit
        RZ      ;Return if not empty
        ADI     OFFH AND NOT TXMTY ;Flag the empty state
        RET
;
; ***** DELETED CODE *****
;

```

(Continued on next page)

C LANGUAGE TRAINING FROM PLUM HALL

- **C Programming Workshop:** Comprehensive, hands-on C course for programmers.
- **Advanced C Topics Seminar:** Includes Efficiency and Portability; for lead programmers.
- **UNIX Workshop:** Hands-on course in uses of UNIX; for general audience.
- Each is a 5-day course: available in-house or at public sessions.



372 pp, 7 1/2"x10", Price \$25.

● AND AN INTRODUCTORY BOOK ON C LANGUAGE:

Learning to Program in C, by Thomas Plum, teaches C language from the ground up. With or without previous programming experience, anyone acquainted with computers will find a clear description of how C works.

You will find guidelines for writing portable programs that will run on a wide variety of modern computers—micro, mini, and mainframe, with excellent efficiency in all these environments.

PLUM HALL 1 Spruce Ave, Cardiff, NJ 08232
Phone orders: 609-927-3770

☐ Send information
on Plum Hall Seminars
on C and UNIX™

☐ Check
☐ Mastercard ☐ Visa
☐ American Express

Exp. Date _____

Please send me _____ copies of "Learning to Program in C"
(NJ residents add 6%; overseas add \$5 airmail.)

NAME _____

COMPANY _____

ADDRESS _____

CITY _____

STATE _____

ZIP _____

Card No. _____

Signature _____

B2

Circle no. 79 on reader service card.

Cursor Control Listing (Listing continued, text begins on page 32)

```
; The following code has been deleted:
;
;CUSI1: EQU      IOER      ;UNASSIGNED USER CONSOLE (INPUT)
;CUSO1: EQU      IOER      ;UNASSIGNED USER CONSOLE (OUTPUT)
;CUST1: EQU      IOER
;
; ***** COMMENTARY *****
;
; The following California Computer Systems vendor code for a
; typical serial interface to a terminal has been added for
; information only as an example to show where the I/O portion
; of the above UC1: code was derived from. Most CP/M TTY: and
; CRT: physical device drivers will be similar to this code.
;
; I/O DRIVERS FOR THE 8250 ASYNC COMM ELEMENT
; (PHYSICAL DEVICE TTY:)
;
;TTST:  IN        SLSTAT   ;GET 8250 LINE STATUS
;        ANI       RXRDY    ;SEE IF RECEIVE DATA AVAILABLE
;        RZ         ;RETURN IF NOT
;        ADI       OFFH AND NOT RXRDY      ;FLAG THAT DATA IS
;        RET                          ;AVAILABLE
;
;TTYIN: CALL      TTST      ;GET 8250 LINE STATUS
;        JZ        TTYIN    ;LOOP UNTIL DATA IS IN
;        IN        SDATA    ;READ THE DATA
;        RET
;
;TTOST: IN        SLSTAT   ;GET 8250 LINE STATUS
;        ANI       TXMTY    ;ISOLATE TX BUFFER EMPTY BIT
;        RZ         ;RETURN IF NOT EMPTY
;        ADI       OFFH AND NOT TXMTY      ;FLAG THE EMPTY STATE
;        RET
;
;TTYOUT:CALL      TTOST     ;GET 8250 LINE STATUS
;        JZ        TTYOUT   ;WAIT UNTIL ONE OF THE REGISTERS EMPTIES
;        MOV       A,C      ;MOVE THE DATA OVER
;        OUT       SDATA    ;OUTPUT THE DATA
;        RET
;
; ***** END OF LISTING *****
```

End Listing

GREAT DRIVES. GREAT PRICES.

Floppy Disk Services offers both.

We supply the industry with high quality disk drives and peripherals. And our prices are some of the best you'll see. Floppy Disk Services is a contracted dealer for Siemens, Tandon and Shugart. Starting with their drives, we design complete packages to give you the system capacity you're looking for and the dependability you need.

We offer add-on drives for IBM, Radio Shack, Heath, Apple and most other microcomputers, all at a significant savings to you. Check the examples below and you'll see what we mean.

Apple II Add on drives	\$300.00
Apple 8 inch controller	315.00
Apple dual 8 inch system w/controller	1165.00
FDD-100-5b 'floppy' exact HEATH add on	215.00
FDD-200-5 double sided 40 track drive	250.00
SA 455 half hgt DS/DD 48TPI ...	\$295.00 .. 2 for \$245.00 each
SA 465 half hgt DS/DD 96TPI ...	\$350.00 .. 2 for \$295.00 each
FDD-221-5 5ms step 80 track DD/DS	330.00
FDD-100-8d 8" single side DD drive	\$240. 2 / \$225. each
FDD-200-8p Double sided 8" drive	\$340. 2 / \$325. each
TM-100-2 (IBM)	265.00
SA-860 DS/DD Half Hgt 8"	\$495.00 2 for \$445.00 each
System packages available for all drives	
Dual 8 inch system with EVERYTHING	750.00*
Dual double sided 8 inch system	950.00*
Single 5 1/4 Heath or MOD I Add on w/case	265.00*
Dual 5 1/4 Heath or MOD I	505.00*
10mb Hard Disk for any computer	2300.00*
Magnolia controller, allows any combo 8 and	
5 1/4 inch drives to be added to your H88 or H89	525.00

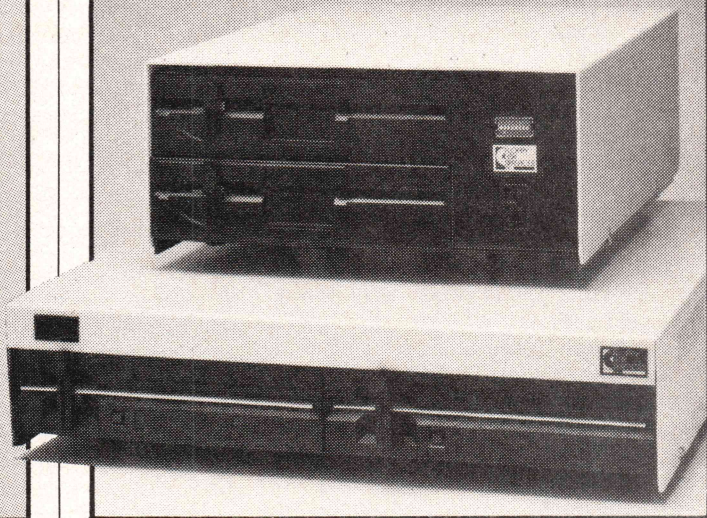
* 8 inch systems are fully assembled and tested, however the drives are shipped separately from the case to comply with UPS weight restrictions. All 5 1/4 inch systems come assembled and tested.

Prices and specs subject to change without notice.

There's much more. If you don't see what you want, give us a call between 9 am and 5 pm (ET). Chances are we'll have what you need at your price.

Featuring the Shugart Thinline Series.

Floppy Disk Services carries the new Shugart Thinline Dual 8 Inch Drive (Model SA-860). It's a double-density drive, with dual head for a storage capacity of 1.25 megabytes per drive. The system includes our custom cabinets, comes fully assembled and tested, and uses the Power One CP-206 supply, the standard of the industry. Special options available: external drive select using DIP switches, and cable connector on the rear of the cabinet for ease in connecting to your system. Floppy Disk Services designs and builds these systems from the ground up to maximize efficiency and minimize space requirements. Available with Tandon Drives also.



Custom Enclosures. At Floppy Disk Services, we also sell disk drive enclosures, designed by our own experts to be functional and attractive. And our quantity pricing is so competitive, we invite dealers and group purchasers to call.

Thinline/Half Height Specials. Floppy Disk Services has just contracted with Shugart, and we're carrying their half-height 5 1/4 & 8" systems. If you would like some unbelievable prices on SA-455, SA-465, SA-860's and others, call today! We are legitimate contracted dealers—no middle-man.

Free Catalogue. If you'd like to receive our Catalogue of Disk Drives and Peripherals, just call or write — we'll mail your copy immediately. And if you want to talk with an expert about getting more out of your system, we'll be happy to help.

Toll Free Order Line (800) 223-0306
Tech Help or Info. (609) 799-4440

**FLOPPY
DISK
SERVICESTM
INC.**

741 Alexander Road Princeton, NJ 08540

Due to production deadlines, prices in this ad are 2 months old, so we encourage you to call us for current prices and new product info.

PAYMENT POLICY — We accept MasterCard, VISA, personal checks & MO. We reserve the right to wait 10 working days for personal checks to clear your bank before we ship. All shipping standard UPS rates plus shipping & handling. NJ residents must add 6% tax.

Dysan's Digital Diagnostic Diskette

Dilemma: You've been seeing more flexible disk errors lately, especially when swapping disks between your drives. But they seem to work properly most of the time, so you struggle on, tolerating the occasional loss of your data, frustration, and wasted time. After all, you're not sure anything is wrong with the drives; to find out, you'd have to remove one from the system and ship it off to a stranger who probably makes money five times as fast as you do. If the problem persisted after the drive came back, would you send another one off to be checked?

Deliverance: Now, for less than what you'd probably spend having one drive professionally checked, you can equip yourself to verify the operation of all your flexible disk drives, anytime you suspect a problem, without disassembling or altering your system! But you'll have to begin now, before you have serious operating problems — partly because you'll have to customize a public domain program to fit the quirks of your terminal and disk controller, but mainly because you'll want to have a "before" picture of your drives' operation to help you recognize any real problems that may appear later.

You thought a drive exerciser and dual-channel triggered sweep oscilloscope were essential for flexible drive alignment? Well, the Dysan Corporation, which makes the Analog Alignment Diskette™ for scope alignment, now offers a Digital Diagnostic Diskette™. Using your own computer system and the program described in this article, you can verify your drives' alignment to within one-half milli-inch of the expensive analog method. Plus, you can check the spindle bearings and clamping mechanism, verify spindle speed, check positioner hysteresis (worn lead screw, bad stepper motor, etc.), verify read/write head azimuth, and check index-hold timing, all under your own operating conditions.

by Loren Amelang

Loren Amelang, Box 24, Philo, CA 95466-0024.

"Analog Alignment Diskette," "AAD," "Digital Diagnostic Diskette," and "DDD" are trademarks, and "Dysan" is a registered trademark, of Dysan Corporation.

What the DDD Is

The Digital Diagnostic Diskette (DDD™) looks like any other floppy disk. It has ordinary formatting, and conventional data are recorded on it. Figure 1 (page 42) shows a small sample of disk data patterns to scale. You could probably run your disk copy program and duplicate it. But your copy would not be another DDD! The digital alignment concept depends upon having the data fields of the DDD carefully displaced from their proper location or azimuth by known amounts.

For example, let's look at the centering test. Three "alternate offset" tracks on the DDD are offset 7, 8, and 9 milli-inches from the official center of the track (see Figure 2, page 42). (All values given are for 48 track-per-inch drives; if you have 96 or 100 tpi drives, tolerances are tighter and thus offsets are smaller.) All odd-numbered sectors are displaced toward the center of the disk, and even-numbered sectors toward the outside edge. Your head gap is designed to pick up a 12 milli-inch swath of data, so if it is perfectly aligned but the data are offset 7 milli-inches, it "sees" only 5 milli-inches of the data. In analog terms, the amplitude of the signal is reduced drastically; in digital terms, either your drive's electronics can make sense of the weak signal, or you get a read error. As you move to the 9 milli-inch offset track, the width of the recorded data still under a properly positioned head is reduced to 3 milli-inches; if your spindle bearings have .003 inches of play, the signal may disappear entirely at certain points. If your head is dirty, and the accumulated oxide holds it away from proper contact with the media, the already reduced signal amplitude will be cut still further. On the other hand, if your system can reliably read data that are offset 9 milli-inches in either direction, you can be pretty sure that your drive is mechanically sound, its head is reasonably aligned, and its electronics are not being swamped by stray electrical or magnetic fields.

If you want to pin down more exactly the radial alignment of your read/write head, the DDD has "progressive offset" tracks (Figure 3, page 44). Here sector one is offset 1 milli-inch toward the spindle, sector two 1 milli-inch away, sector three 2 milli-inches toward the spindle, and so on out to 13 milli-inches in either direction. (This applies to Revision B; Revision A DDD had ± 10 milli-inch maximum offsets.) You read through the odd-

numbered and then the even-numbered sectors, noting the first sectors that cannot be read. This information can be translated into a graphic display of your effective head width and its position relative to proper alignment. Electrical or magnetic noise in your environment, "dirt" on your head, or improper head loading will narrow the effective head width. A major advantage of the DDD is that you can run this test on six different tracks spaced across the diskette; analog alignment uses only one central track, with the hope the rest will follow.

The "azimuth rotation" track is another progressive track used to check whether your read/write head gap is adversely skewed relative to the data track. Here data are recorded on the radial centerline of the track, but the head azimuth is progressively rotated away from normal (Figure 4, page 46). If read errors occur symmetrically, your effective head azimuth is correct. If your system fails in one direction before the other, you may improve performance by altering your head azimuth toward the earlier angle. Again, a decrease in azimuth range could indicate a dirty head or environmental noise.

Other recorded DDD tracks contain a special formatting pattern for timing the passage of the index hole past your sensor and for detecting how long it takes your head to settle into position after it is loaded. Finally, some tracks are marked "user area," where you can store the program you'll need to read the DDD and display the results. *Don't* try to do this under CP/M! You'll overwrite other parts of the DDD and destroy it! If you are going to make use of the "user area," you'll need a "disk doctor" program that can write to specific tracks of the disk, and you'll need some way to load and run what you've stored without involving CP/M.

How does Dysan Corporation manage to create a disk with precisely misaligned information on it? To begin, they take their premium diskettes and screen them: any alignment diskette must meet high standards for dimensional stability and uniformity of the recording surface. The DDD must be error-free, not only in the normal tracks but "in the cracks" as well. Critical information is going to be recorded completely off track for some of the tests.

Several million dollars went into the computer-controlled, laser-interferometer-calibrated trackwriters and the environ-

mentally standardized room that houses them. Once the blank diskettes have acclimated to the standard environment, they are individually written; the track-writers actually move the write heads to all those different off-track positions. These same machines write the Analog Alignment Diskettes used by manufacturers and calibration services. The same precise standards are met, both for the media and for the recorded information. The only advantage of analog alignment is that it is not limited to resolving the 1 milli-inch steps of the DDD but can get closer to the exact center of track. Of course, the DDD has the advantage of being able to check alignment on six different tracks and may reveal more than one-half milli-inch of variation between the inner and outer track centers.

Using the DDD

Dysan Corporation has been making the DDD for about two years. It was conceived as a way for manufacturers to verify the operation of completed systems quickly, without having to reach any interior test points. Field service personnel have been using it to diagnose drive problems at the customer's location. Recently, the company has been promoting the DDD to end users at various computer shows.

To assist the end user in creating a program that reads the DDD and displays the results of the various tests, the Customer Engineering Division of Dysan Corporation developed the original (version 1.0) DDD program. Designed to run on their "Single Card Computer" with a Hazeltine 1500 terminal, version 1.0 employs direct-memory-access (dma) transfers of read data and requires an 8080 macro assembler for assembly. Luckily for those of us with other types of equipment, constants are clearly defined at the beginning of the code, and the hardware-dependent routines are collected at the end of the code. I found the macro facility was rarely used, so I expanded the few examples into code that ASM can handle. Since my terminal does not have a clear-to-end-of-line function, I inserted a conditional that either prints a string of blanks when it is absolutely essential or allows those who have the function to benefit from its greater speed.

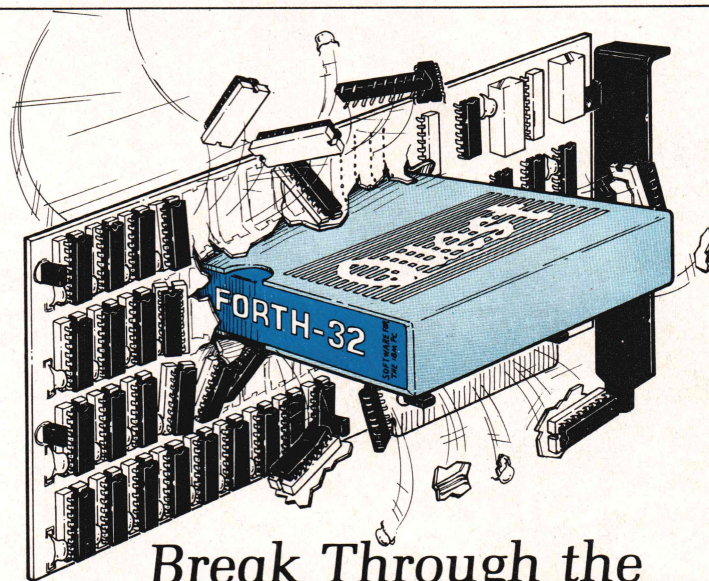
This still leaves a few basic requirements: The program depends heavily on being able to clear your terminal screen and address your cursor. If you have a really dumb terminal, you will have to completely rethink the graphic display parts of the code. The program, as written, commands a Western Digital 1793 floppy disk controller chip; if your system does not use this chip, you will have to understand both the 1793 and your own controller well enough to translate

commands between them. Certain controller chips are not capable of performing all the commands the program uses, so some ingenuity may be required. Different systems have different ways of selecting which drive is to be tested. I have added to version 1.0 a routine to select any of four drives connected to my CCS 2422 controller. You'll have to understand how your system goes about selecting drives. Finally, as should be obvious, you'll need to be able to deal with 8080 assembly language in order to get the DDD program working.

Nothing inherent in the DDD limits it to CP/M or 8080 systems. It is available in models for 8-inch or 5¼-inch drives; single or double sided; single or double density; 48, 96, or 100 spi. Since the DDD

program has been placed in the public domain, it may be used as an example to construct versions for other machines, processors, and operating systems. If you develop a version for a popular computer, I hope you'll make it available on a bulletin board or software exchange.

If you'd like to take the mystery out of your flexible disk system, and you're willing to be guided through some hardware-intensive assembly language, here's your agenda. First, order a DDD. If you have double-sided drives, you'll need the double-sided DDD in order to test both heads. If all your drives have double-density capability, you'll want to test them in this more stringent mode. If some are single density only and your doubles can read single, perhaps you'd



Break Through the 64K Barrier!

Make full use of the memory you have by programming with the **FORTH-32** language. A Complete Development System! Floating Point and other Extensions.

Quest
Quest Research, Inc.

Call toll-free or contact a participating Computerland store.

303 Williams Ave.
Huntsville, AL 35801
(205) 533-9405

Available for DOS 1.0, 1.1, 2.0
FORTH-32 and the Quest logo
are trademarks of
Quest Research, Inc.

800-558-8088

Circle no. 85 on reader service card.

prefer a single DDD for all of them. You'll need to specify the disk size and number of tracks per inch. Call your local Dysan representative, or call the factory at (408) 988-3472 or (800) 551-9000 outside Northern California, and be prepared to spend \$30.00 for a single-sided DDD or \$40.00 for a double-sided DDD, plus shipping and applicable taxes. For more information on DDD

applications, call Dysan's Tech-Line at (408) 734-1624.

Next, you'll need machine-readable source code for the program that reads the DDD and displays the results. DDD11.ASM (or DDD11.AQM in the "squeezed" version) may be available on your local RCPM system. If you have a phone line, modem, XMODEM-Protocol transfer utility, and "Unsqueeze" if

DDD11.AQM is used there, you can copy the program yourself for free. If you must call long distance, be warned that DDD11.ASM is a 47K file. You might be better off downloading it from Micronet's CP/M Interest Group "Access area 1." If you prefer to purchase the code on disk, contact your favorite source for CP/MUG disks or write CP/MUG, 1651 Third Avenue, New York, NY 10028. Should all

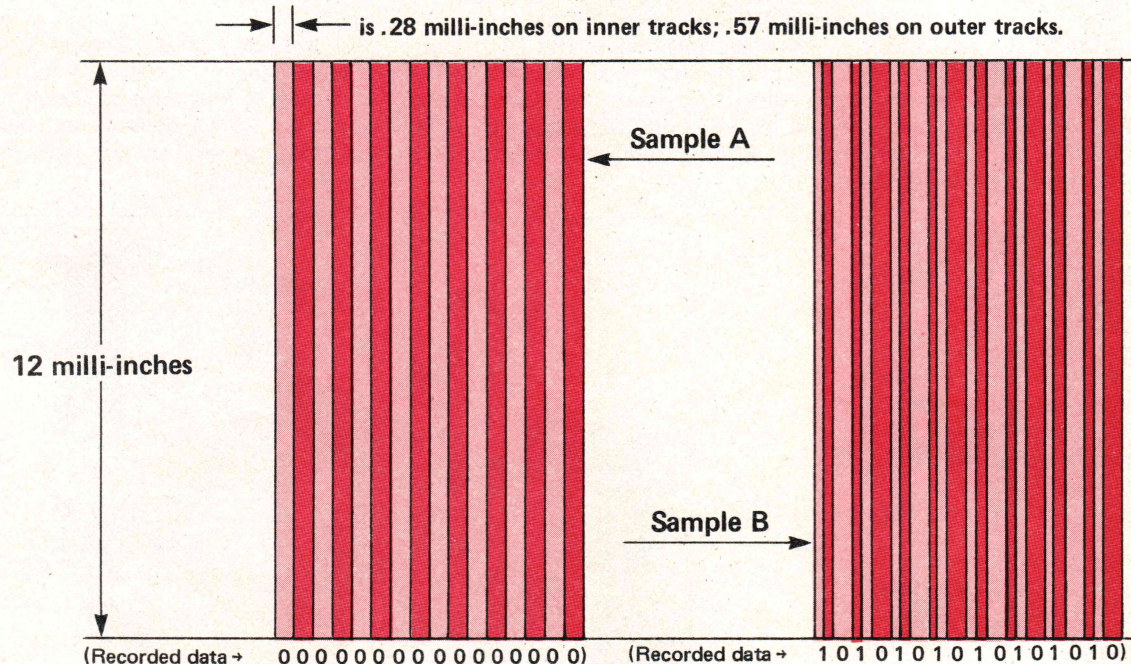


Figure 1.

The actual magnetic pattern on an 8-inch diskette, to scale.

The black lines are *not* on the diskette. There are no prearranged "boxes" to be filled in, only uniform magnetic particles. When you write on the blank diskette, you arrange these particles in alternating stripes of "north" and "south" poles (the light and dark areas in the figure). It is not the polarity itself but the *change* in polarity that is significant. Sample A is a two-byte string of "0" bits, which means that there are no (zero) transitions with a bit space. Sample B alternates between "1" and "0" bits; a transition within a bit space is read as a "1."

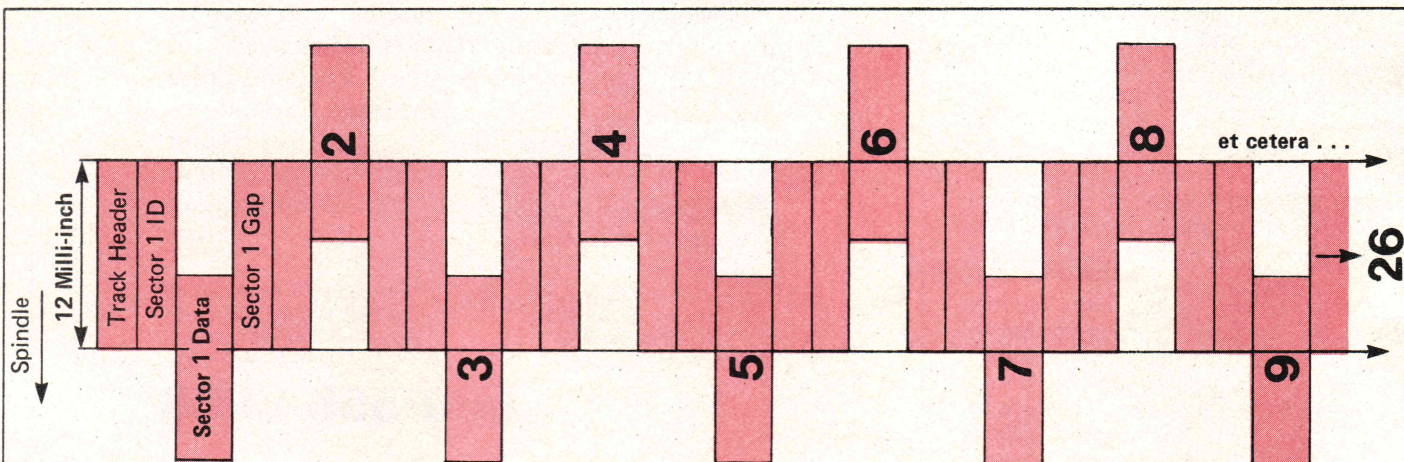


Figure 2.

An alternative offset track.

Each data sector is 128 bytes, while sector IDs are 31 bytes and sector gaps are 29 bytes or less. The track header is 73 bytes. The circular track is shown here stretched out straight for convenience — tracks form complete circles on diskettes.

Don't call her cheap. Call her beautiful.

The Bonnie Blue™

Word Processing System for the IBM Personal Computer

It's obvious what makes her cheap. But what makes Bonnie Blue so beautiful? Bonnie Blue is a totally new word processing system for the IBM Personal Computer. Bonnie Blue really gives you the combination of power and convenience. Ignore the price for a minute, and listen to this.

The Full System

Bonnie Blue is a single program, with three modules, organized as overlays. The heart of the system is the full-screen editor. As we'll show you, it's worth more than the price of the whole system. But the Main Menu also provides access to a complete print formatting module, and the Toolbox. Read on.

Full Screen Editor

Bonnie Blue has a full screen editor, and a complete set of cursor control operations lets you move anywhere on the screen, or scroll the text up or down. All cursor and edit controls are single keys, so there are no sequences to learn. Bonnie Blue can move, copy, delete and paste blocks of text. It can also insert a file into your text, or extract a block of text onto a file. You can find strings and insert, delete or replace them, either individually or over a range of lines.

Bonnie Blue does not force you to go back and forth between different "modes"; creating and editing text are part of the same process. You can turn on or off automatic wraparound, centering, justification, and addition of underline or other attributes to the text. By turning these off, you quickly turn Bonnie Blue from an excellent word processor into a superb text editor, with special features for programmers, such as auto indent.

You can remap any key to any other key; this is done within the program, and saves you the cost of the whole package by itself. You can put any displayable

character into your text, with lines up to 220 columns wide.

Bonnie Blue makes unique use of the display attributes (underline, boldface, etc.). You can "paint" these attributes onto your text, and then, when printed, they can be mapped to any single feature or combination of features which your printer can provide. For example, you might want text which is underlined on the screen to be underlined when printed, but you could just as easily make it underlined italics, or reverse video might mean boldface and wide text. The dot command .USAGE can change what the attributes mean during the printing process. All files produced by Bonnie Blue are DOS standard.

Print Formatter

The print formatting module is coordinated with the editor.

Our first print module supports all the features of the Epson MX printers with Grafrax Plus. By the time this ad appears, we will be supporting other popular dot matrix and letter quality printers as well.

The print module can print multiple

copies, controlled from a "mailing list" file, with insertion of data from the file into the document, for the production of form letters.

The Tool Box

Bonnie Blue has a full set of "software tools" which bring Unix-type productivity to text manipulation. There are tools to include and concatenate files, count words, sort, and transliterate characters.

We plan continued expansion of the Tool Box including a list manager, telecommunications, and aids for the C programmer, including our version of lint.

Can There Be More?

Bonnie Blue is not copy protected, so it's great for a hard disk environment. The manual is large and thorough. It runs incredibly fast. And there are many other features too numerous to mention here.

If you have an IBM PC, 128k memory, and an Epson or Epson-compatible printer, order today. Or send for our brochure. We are developing versions for other 8086 and 80186 based systems.

Yes. Send me the Bonnie Blue Word Processing System. Enclosed is my payment of \$50 (NY State residents add 7% sales tax). Mail coupon and payment to:

Bonnie Blue Software, P.O. Box 536, Liverpool, NY 13088

☐ Check enclosed ☐ VISA ☐ MasterCard Sorry, no COD.

Credit Card No. _____ Expires _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

Company _____

☐ Please send literature. I have a _____

Only \$50

IBM Personal Computer is a trademark of IBM Corp.
Epson is a trademark of Epson America, Inc.

CA 95466-0024, and enclose \$10.00. Listing One (page 48) contains a slightly abbreviated version of DDD11.ASM, called DDDCompact.ASM.

Finally, you'll need some information about your hardware. Check your terminal's manual to see if it gives clear-screen and addressable cursor-control commands. See if your disk controller manual gives control port addresses, command formats, status codes, and flow charts of command execution. If the information isn't there, ask the manufacturer if more detailed programming information is available. While a source listing of your CP/M BIOS would probably reveal what you'll need to know about the disk controller, this will be more difficult to decipher. If your hardware is based on an LSI controller chip, the chip manufacturer's data sheets will be very helpful. Save yourself frustration by collecting your hardware information before you begin — while you wait for the DDD disk and DDD11.ASM to arrive.

Customizing DDD11.ASM for Your System

Once you've chosen which DDD format you need, collected your hardware manuals, and are seated at your keyboard with DDD11.ASM in your editor, we can begin. You've read the headers down to "Program Source Code." The first instruction to the assembler is "wboot equ 0" — this is nearly always true. If you happen to be dealing with a nonstandard CP/M system, you probably are well aware of the fact; you've needed special versions of programs from day one with your system and will have to modify my recommendations as we proceed.

Next come the "Conditionals." Here you must decide how committed you are to good programming style. The quick way to customize a program is to step through the code and change only a few magic numbers until it works. The right way, which will pay off handsomely if you should want to give the program to a

friend or need to understand what you did come next year, is to create conditional assembly options here for both your terminal and your disk controller. Whenever you need to alter a bit of code in the body of the program, insert "if My____," enter your changes, and then write "endif." By setting "My____" equ "true" in the "Conditionals" section, you insert all your changes into the assembled code. If your friend wants the original code back, he or she can remove all your changes by setting "My____" equ "false." Unless you happen to have one of the existing conditional systems, set these conditionals equ "false," add similar lines for your terminal and controller, and set yours equ "true." Set "lineclr" true or false according to whether your terminal can clear-to-end-of-line with a single command.

The "Macro Call" section has been commented out, and all examples of its use have been expanded by hand, so that you can use a simple assembler like ASM.

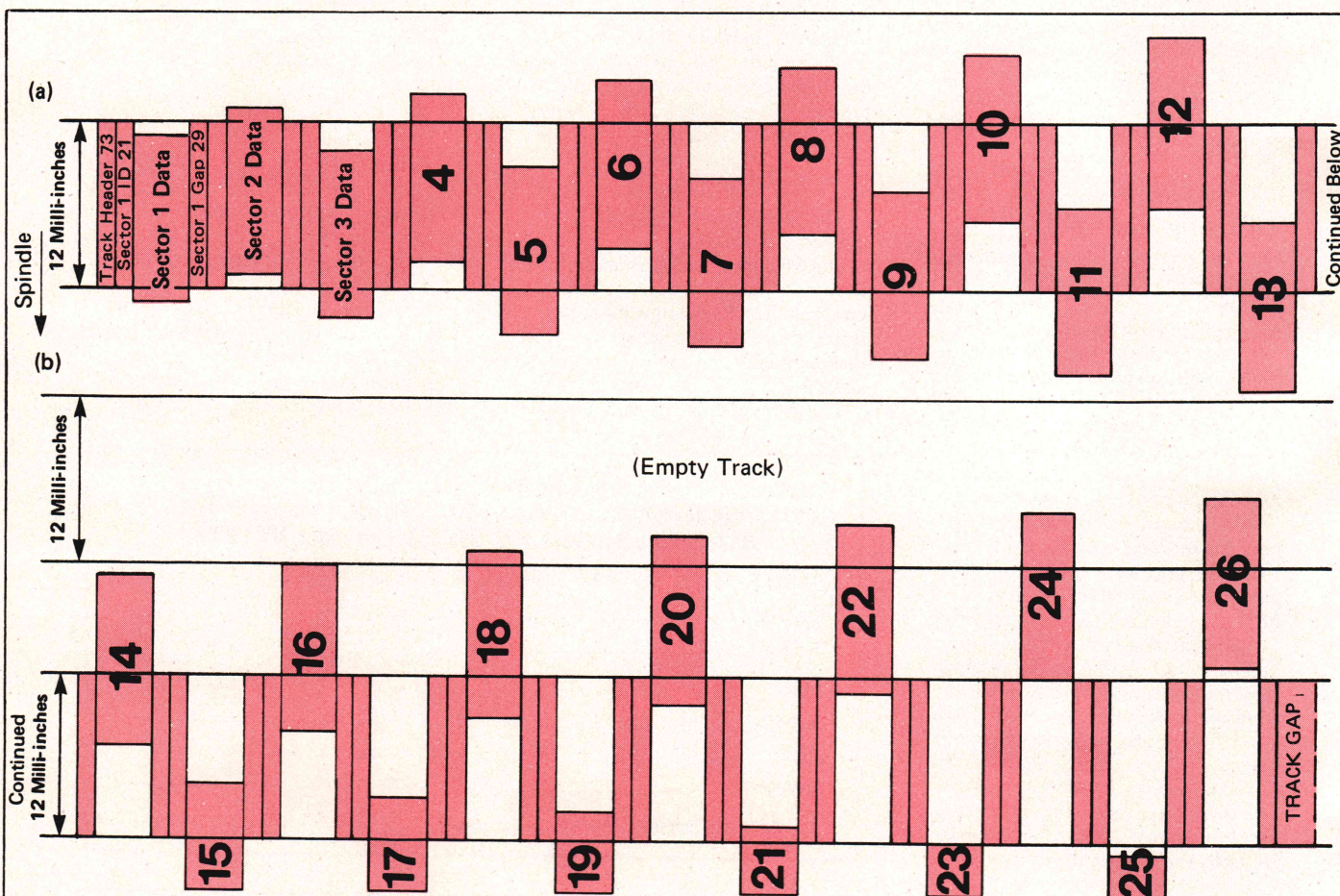
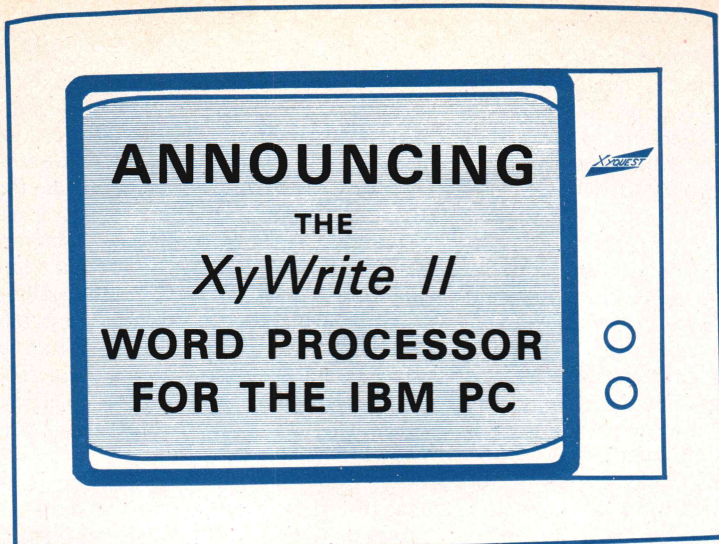


Figure 3.

A progressive offset track.

Vertical scale is 75 to 150 times horizontal, depending on the track number. The entire track is displayed here, split into two parts: sectors 14 through 26 (at (b), the bottom part of the figure) are displayed with a neighboring empty track in order to show how the offset sectors begin to bleed out into the adjacent empty tracks; sectors 1 through 13, at (a), are shown without neighboring tracks.



from



We hereby make you an UNBELIEVABLE OFFER:

For only \$7.00 (\$5.00 for the disk, and \$2.00 for handling), we will send to you a DEMO DISK of XyWrite II, THE ONLY WORD PROCESSOR to take advantage of the SPEED and POWER of the IBM PC.

We believe in our program so we want to make it easy for you to try it. We want you to compare XyWrite II against the rest. You will quickly see the lightning responsiveness of the program: it'll take your breath away.

FASTEST PROGRAM ON THE MARKET

SIMULTANEOUS MULTIPLE FILE ACCESS

SPLIT SCREEN WRITING (HORIZONTAL OR VERTICAL)

SPLIT SCREEN EDITING (MOVE TEXT BACK & FORTH)

HORIZONTAL SCROLLING

EDIT OR WRITE WHILE PRINTING

INTEGRAL MATH PACKAGE

COLUMNAR POSITIONING

INDEXING, FOOTNOTES, HEADERS/FOOTERS

and MUCH, MUCH MORE

Find out what real speed power is on your IBM PC. Compare XyWrite II with WordStar, EasyWriter 1.1 and II, Benchmark 3.0, PowerText, Volkswriter, FinalWord, Select, WordPlus, PerfectWriter - or any of the others.

You have nothing to lose. The high quality diskette is yours to keep. When you decide to buy XyWrite II, you receive a \$7.00 discount off the \$195.00 list price. Call or write us today at (617) 275-4439. Ask for John.

WordStar is a registered trademark of MicroPro International. Benchmark is a registered trademark of Metasoft Corp. PowerText is a registered trademark of Beaman Porter, Inc. Volkswriter is a registered trademark of Lifetree Software, Inc. FinalWord is a registered trademark of Mark of the Unicorn. Select is a registered trademark of Select Information Systems. WordPlus is a registered trademark of Professional Software, Inc. PerfectWriter is a registered trademark of Perfect Software. EasyWriter is a registered trademark of Information Unlimited Software, Inc. IBM and IBM PC are registered trademarks of International Business Machines Corporation.

**XyQuest Inc.
P.O. Box 372
Bedford, MA 01730**

Ignore it. Under "Console Functions" comes your first chance to say "if Myterminal." Set as many of the variables as your terminal uses to the values given in your manual. Don't forget the "endif." Check the ASCII code equates, especially backspace, for compatibility with your terminal.

In "Floppy Controller Port Equates," you tell the program where to send commands for the disk controller. Set up a conditional for your controller; you can probably copy these addresses directly from your BIOS source, if you have it. Note how the base address is given in absolute numbers, and all the other addresses are given as offsets from the base. Most systems with a 1793 controller chip will have these same offsets from whatever base address the system uses.

In the "if Dysan" conditional, "dma" is a port address where commands to the direct memory access controller are sent. Each time this controller is asked to read the disk, it is sent an address in memory at which to begin storing what is read. It reads the disk and puts the data directly into memory, while the CPU does other tasks, by using the slivers of time between CPU memory accesses. In my CCS conditional, "dma" is a memory address where a pointer to a memory buffer is stored. In this non-dma or programmed I/O transfer system, after telling the controller to read the disk, the CPU must be ready to grab each byte as it is read and to move it into memory before the next byte is ready. The CPU looks at "dma" to find out where in memory to begin storing the read data. Obviously, the CPU cannot do anything else while it is busy moving in data; in fact, to keep up with a double-density 8-inch disk it must run at least at 4 MHz.

"Floppy Controller Commands" is another section you might copy from your BIOS source. Command "sdma" tells the Dysan controller to begin transferring data; if you have a dma-type controller, it should have a similar command.

Variable "fdelay" is used in the "delay" routine, where time is being carefully counted. Instead of issuing a command to the controller and checking to see if it is processed, this routine dispenses the delay time in known chunks of 50 microseconds. If you don't have a 4 MHz Z80, and you want the timing routines to work, "fdelay" and the "delay" routine will have to be juggled until the loop takes 50 microseconds. "Indxbyt," "busybyt," and "drqbyt" are masks that select the appropriate bit from the status register of the controller chip. For example, if the status register is read, and the logical "and" function is done with "busybyt," a 0 will result unless the bit signifying "busy" is set high. A bit-by-bit explanation of status register contents should be available in your controller chip data sheet. The other variables in this section go directly to the controller chip and can be determined from the manufacturer's data sheet, if not from your BIOS source.

In "DDD Diskette Variables," the total number of tracks and the number of sectors per track in your chosen format are set. Note that several other parameters are defined in terms of sectors and need not be changed. "Ref" is a magic number that will be discussed later under "Index Timing." Ignore it for now. Those of you with nonstandard CP/M systems will have to change the "org" address. Otherwise we're already down to the actual program!

Step on down to "Clear Console Display" — the code in between should run unchanged on any CP/M system. If your terminal, like the Hazeltine, needs an extra lead-in byte ahead of the actual clear-screen command, you'll need to put in an "if Myterminal" here. "Clear Display Line" has two conditionals: one for terminals with a clear-to-end-of-line command, which is active if "lineclr equ true," and another that prints a string of blanks for terminals without this feature. If you've set "lineclr" true, and if your terminal needs a lead-in byte or other special treatment, insert an "if Mytermi-

nal AND lineclr" as I have for the Hazeltine. Again, in "Print String of Text," insert a conditional under "text1" for lead-in type terminals. If your terminal should need the row position sent before the column position, accomplish that here by incrementing the HL register pair twice after the position cursor command (to get the row position) and then decrementing HL to get back to the column position. This code section assumes that the upper left-hand corner of your screen is called 0,0. If your terminal places 0,0 somewhere else, you'll have to add a constant bias to the coordinates before you send them.

"Index Timing" may require major changes. In the Dysan dma system, the command "read address" is sent to the dma controller, "start dma" is sent, and the CPU is set to counting the time until the controller is finished doing its direct memory access. If you have a dma-type controller, you'll use this scheme as it is. If you have a programmed I/O controller, the CPU cannot transfer data and count time simultaneously. My solution, in the CCS conditional, is to send the "read address" command and immediately set the CPU to counting time until "drq" comes true. When the first byte of address data is ready to be transferred to memory, the disk controller sets "drq" true, telling the CPU to "come and get it." The CPU immediately determines (lhld dma) where to put the byte, reads it from the data port, and moves it to memory. It increments the memory address and goes after the next byte. "Read address" reads only six bytes of data, so I wrote six reads inline.

Since the non-dma CPU only counts time before the data transfer, the time counting loop after that code is for the Dysan controller only. But because the Dysan dma controller counts both the time from the "read address" command to the first data byte and the time needed to transfer six bytes of data, it will get a longer time count. This is where "ref"

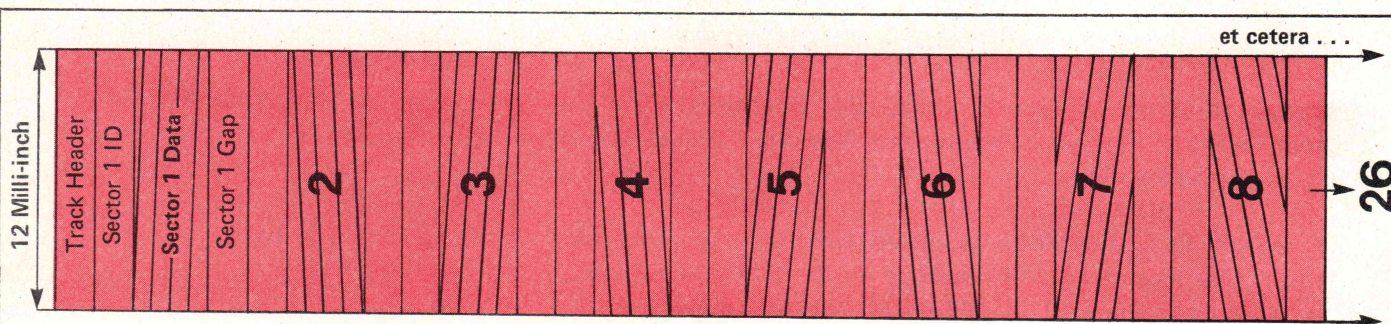


Figure 4.

An azimuth rotation track.

Data sectors, sector IDs, sector gaps and track headers take the same number of bytes as in Figure 2. Azimuth rotation is *highly exaggerated!* Actual rotation is from 18 to 42 minutes — less than a degree. Again, the track is drawn straight for convenience.

comes in. "Ref" is a magic number or, more bluntly, a fudge factor, which makes the final count read however we want it. Since "ref" is subtracted from the total time count, the non-dma controller will require a smaller "ref" value than the dma controller. Dysan's value is experimentally set with their Analog Alignment Diskette, whose track has a standardized 200 microsecond time from index hole to address mark. My value of 24EH is estimated to make my DDD also give about 200 microseconds on my drives — hardly a precise method, but a relative measure is better than no measure at all.

"Index Timing" is the first section to introduce the problem of time counting. You've undoubtedly noticed that many of the instructions are followed by numbers in the comment field. These numbers give the execution times in microseconds for those instructions on a 4 MHz Z80. If your CPU is clocked at 2 MHz, the instructions will take twice as long. But disk reads of data happen at the speed of the moving diskette, regardless of the CPU clock. If you intend to properly adapt this routine to a different clock rate, you will have to manipulate not only the timings here, but also the delay provided by "delay" under "FDC Delay to Process Commands" and the loop count "fdelay" in "Floppy Controller Commands." Unless you enjoy mathematical puzzles, I suggest you join me in a cut-and-try fudge. In "Index to Index Timing" and its "RPM Timing Loop," no disk reads are involved, only time counting between index pulses. Here a 2 MHz processor should get exactly half the proper time, and a mathematical correction is reasonably simple. Translate the instruction times into their corresponding values at your clock speed, and set fewer loop passes or adjust the trailing "inx d" instructions to give the required 100 microsecond loop.

"Drive Select" is your chance to be original. The version 1.0 option simply selects one drive only. My CCS option can select one of four drives connected to a CCS 2422 controller, but it does not deal with double-sided drives. The code down to "jmp select" should be pretty universal. After that, you'll have to determine from your manual (or crib from your BIOS source) the codes needed to select drives under your controller and then invent a way to turn "A,B,C, or D" into whatever codes you need. If you need to select sides on a double-sided drive, duplicate the top part of the CCS option then display a "Which Side?" message, translate the response into the proper code, and send it to the proper port.

In the "Read Sector Routine," we again have to separate dma controllers from non-dma controllers. With a dma controller, just issue the go command and

Our new Optimizing = C86™ C Compiler controls Charlie...

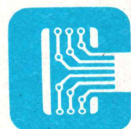
Like a Puppet on a String!

Take control of Charlie with our new Optimizing = C86™ C Compiler...now available for the IBM PC-DOS and the MS-DOS operating systems. Take the upper hand with:



- Significantly faster and tighter code
- Standard object module format, linkable with the MS-DOS linker
- Compile time switch for big machine addressing (Programs can be as large as the memory you can afford!)
- Compile time switch to produce in-line 8087 code. (We still provide our floating point package, too!)
- Significant performance increases for our I/O library
- MS-DOS version 2.00 I/O library
- Many additional library functions
- Expanded manual with numerous examples
- Librarian for maintenance of MS/DOS libraries
- Optional assembly source output (It's not NECESSARY to produce assembly source, but it CAN be assembled using MASM!)

Pull Charlie's strings with our new Optimizing = C86™ C Compiler. See your local dealer or call Computer Innovations for information and to order the C86 Compiler Diskette and Manual...still only \$395.00!



Computer Innovations
10 Mechanic Street
Suite J-506
Red Bank, N.J. 07701
(201) 530-0995

**"They Say It All...
We Do It All!"**

Visa and MasterCard accepted.

C86 is a trademark of Computer Innovations, Inc. CPM-86 and MPM-86 are trademarks of Digital Research. MS-DOS is a trademark of Microsoft. PC-DOS is a trademark of International Business Machines.

Circle no. 23 on reader service card.

check to see when the status register says all the data are transferred. With a programmed I/O controller, you'll have to tell the CPU every step to take to get the data into memory. Under "read1" in the CCS option, I loaded HL with the memory destination of the first byte and initialized a loop counter. Each loop through the data transfer code reads four bytes, so to read an 8-inch single-density sector takes 32 (20H) loops. Double-density 8-inch sectors require 64 (40H) loops. This routine can probably be adapted from the disk read routine in your BIOS. You could call the BIOS routine, but that would make your DDD program nonportable.

"Program Disk Controller" originally output a series of commands to a dma controller, reading them in serial form from the table that followed. If you have a dma controller that needs to be "primed" with a string of commands, create your own conditional table and adjust "Bytes count" for the number of bytes you need to send. Your manual should explain the procedure; otherwise, find a data sheet on the actual chip used or copy from your BIOS. Even a non-dma controller needs some priming. Mine will look at memory address "dma" (remember "Floppy Controller Port Equates"?) and expect to

find the address of "secbuf," a memory buffer to move data to. The other item needed to prime it for a disk read is the setting of "autowait," a function of the CCS 2422 controller card that puts my CPU into a wait state from the time it sends a disk read command until the first byte of data is ready. The catch is that to set "autowait" I have to write to the same port that does disk select; thus I save the bits I used to select the current disk and sent them here with a double function.

"Delay" has already been dealt with (or fudged), so unless you'd like to customize some of the console displays or messages, we've reached the end of the source code! Run it through your assembler and clean up any trivial errors, and the program should be yours. To make it a useful tool, you'll need to create a "before" picture of your drives' operation. Run all the tests on all your drives and write down the results. There are no "correct" results for these tests; however, you'd like to be able to read as many sectors as possible and to have your readings symmetrical around the center of a properly aligned track. If you record a baseline while your drives are working, you'll be able to identify any subsequent changes as possible

causes of new malfunctions.

Finally, be aware that flexible diskettes, including the DDD, change size with temperature and humidity. A change of 18°F can move the outer tracks of an 8-inch disk up to 1 milli-inch. On an extremely humid day, the outer tracks may "grow" another 3 milli-inches. If your drives are in a poorly ventilated enclosure where the temperature rises to 125°F, that alone can move tracks 7 milli-inches out of alignment. Run your baseline tests under normal conditions of operation after your system is well warmed up, and be sure the DDD has had time to acclimate to the room and to the temperature inside the drive you're testing. Proper drive alignment once the drives are packed into your enclosure may be considerably different from alignment in the open on a test bench.

DDJ

(Listing begins below)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 199.

Diagnostic Diskette Listing (Text begins on page 40)

```

1:
2: ;      DIGITAL DIAGNOSTIC DISKETTE tm APPLICATION PROGRAM
3:
4: ;      BY LOREN AMELANG, BOX 24, PHILO, CA      95466 0024
5:
6: ;      THIS CODE EXCERPTED FROM VERSION 1.1  (DDD11.ASM)
7:
8: ;      (THE EXCERPT DOES NOT INCLUDE STEPPER HYSTERESIS
9: ;      OR INDEX TIMING CHECKS...)
10:
11: ;      This document contains information developed by
12: ;      Dysan Corporation (Dysan) and is furnished for
13: ;      information only.
14:
15: ;      Dysan makes no warranty or representation (expressed
16: ;      or implied) with respect to the accuracy, completeness
17: ;      or usefulness of the information contained herein.
18: ;      Further Dysan assumes no responsibility for liability
19: ;      or damage of any kind which may result from the use
20: ;      of the information contained herein.
21:
22: ;      THIS PROGRAM IS IN THE PUBLIC DOMAIN, AND MAY BE
23: ;      FREELY COPIED AND DISTRIBUTED
24:
25: ;
26: ;      *****
27: ;
28: ;      USAGE:
29: ;
30: ;      To use this program, a digital alignment disk is required.
31: ;      You may purchase the required "Digital Diagnostic Diskette" (tm)
32: ;      directly from Dysan Corporation. Available DDD(tm) models are:
33: ;
34: ;      SIDES:.....SINGLE SIDED.....DOUBLE SIDED

```

(Continued on page 50)



8087 Number Cruncher

for CompuPro[®], NEC and
other microcomputers.*

* North Star and Zenith versions by 4th quarter, 1983.

"This will be a boon to those of you who wish to take advantage of **8087** for fast floating-point math but don't want to trash your present CPU board..." —Ray Duncan, *DR. DOBB'S JOURNAL*
"... the result (from the **Hudson 8087 Support Board**) is well worthwhile." —Jerry Pournelle, *BYTE*

Perform math operations at the speed of a mainframe computer. The **Hudson 8087 Support Board™** plugs right on your processor board, permitting you to perform numeric operations **UP TO 300 TIMES FASTER!** Installation takes less than five minutes—requires no software modification.

Additionally, the **8087** is being supported by virtually every major language house, including Digital Research and Microsoft.

NOW 8-BIT LANGUAGE SUPPORT FOR THE 8087 SUPPORT BOARD

Typical double precision (64-bit) benchmarks:**

Operation (5000 iterations)	FORTRAN-80	FORTRAN-87/BASIC-87	
		(5 MHz)	(8 MHz)
multiplication	32 sec.	2.4 sec.	1.5 sec.
division	62	2.5	1.6
sine or cosine	380	3.1	2.0
logarithm	390	2.6	1.7
square root	500	1.7	1.1

FORTRAN-87™ and BASIC-87™ put the **8087** to work for you using Microsoft's popular BASIC and FORTRAN compilers. Use your regular CP/M 2.2[®] and FORTRAN-80/BASIC-80 compilers on your CompuPro, Zenith, or North Star dual-processor systems interfaced with FORTRAN-87/BASIC-87 and utilize the floating point power of the **8087 SUPPORT BOARD**, increasing your math processing up to several hundred fold.

8087 SUPPORT BOARD: \$495.00

(Includes 5 MHz 8087 processor; 8 MHz 8087 \$95 extra)

FORTRAN-87: \$200.00 BASIC-87: \$200.00

Postage and handling included. Prepaid or C.O.D. orders accepted. Give street address for UPS delivery, and allow 3-6 weeks. Dealer inquiries welcomed. Specifications subject to change without notice.

Hudson Marketing

** Results were obtained on a CompuPro 8/16 computer with an 8085 running at 6 MHz and an 8088 running at 5 MHz and 8 MHz respectively.

829 Rosewood Court, Walnut Creek, CA 94596 — (415) 944-9680

Circle no. 48 on reader service card.

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```

35: ; DENSITY:.....SINGLE(FM) DOUBLE(MFM)...SINGLE(FM) DOUBLE(MFM)
36: ; SIZE & TPI:
37: ; 8" 48 TPI 800-100 800-200 800-300 800-400
38: ; 5.25" 48 TPI 500-100 500-200 500-300 500-400
39: ; 5.25" 96 TPI 506-100 506-200 506-300 506-400
40: ; 5.25" 100 TPI CALL FOR AVAILABILITY CALL FOR AVAILABILITY
41: ;
42: ;
43: ; PRICES: Single Sided - $30.00 Double Sided - $40.00
44: ; Plus appropriate sales taxes and shipping costs
45: ; Prices and models subject to change without notice
46: ;
47: ;
48: ; ORDERING: Contact your local Dysan sales Rep. or call
49: ; Inside Northern California: (408) 988-3472
50: ; Outside Northern California: (800) 551-9000
51: ;
52: ;
53: ; TECHNICAL INFORMATION: For information regarding DDD(tm)
54: ; applications contact Dysan's
55: ; CE Division TECH-LINE: (408)734-1624
56: ;
57: ;
58: ; *****
59: ;
60: ; PROGRAM SOURCE CODE:
61: ;
62: ;
63: wboot equ 0 ; warm boot address
64: ;
65: bdos equ 5 ; jump address for bdos calls
66: ;
67: ;
68: *****
69: *** "CONDITIONALS" ***
70: *****
71: ;
72: ; Choose the appropriate assemble time options for your system,
73: ; or use this space to add new system options.
74: ;
75: true equ 0ffffh
76: false equ not true
77: ;
78: Dysan equ false ;set this true for Dysan controller
79: CCS equ true ;set this true for CCS 2422 controller
80: ; programmed i/o transfer, etc.
81: lineclr equ false ;set this true if your console has
82: ; a clear-to-end-of-line command
83: Hazel equ false ;set this true for Hazeltine 1500 terminal
84: ; (requiring "lead-in" to commands)
85: Loren equ true ;set this true for Loren's homebrew terminal...
86: ;
87: ;ADD CONDITIONAL OPTIONS FOR YOUR ENVIRONMENT HERE...
88: ;
89: ;
90: ;
91: ;
92: *****
93: *** "CONSOLE FUNCTIONS" ***
94: *****
95: ;
96: ; The followings are direct console commands
97: ; which can be altered for most consoles.
98: ;
99: ; note:
100: ; The text print routine must be altered
101: ; if "Row" is sent before "Column" or
102: ; bias is needed.
103: ;

```



```

104:      if      Hazel      ;version 1.0
105: add$cur equ 17          ; address cursor command
106: clr$co  equ 28          ; clear to foreground spaces
107: lead$in equ 126        ; console lead-in
108: end$ln  equ 15         ; clear to end of line
109:      endif
110:
111:      if      Loren      ;version 1.1
112: add$cur equ 01          ; address cursor command
113: clr$co  equ 05         ; clear to foreground spaces
114:      endif
115:
116: cr      equ 0dh          ; carriage return
117: lf      equ 0ah          ; line feed
118: esc     equ 1Bh         ; escape
119: bs      equ 8           ; back space
120: bell    equ 7           ; audio alert
121:
122:
123: *****
124: *** "FLOPPY CONTROLLER PORT EQUATES" ***
125: *****
126:
127: ;      Change the following Port assignments for
128: ;      Your system.
129: ;
130: ;      "FDC" Acronym Floppy Disk Controller
131: ;
132: ;      FDC port assignments:
133: ;
134:      if      Dyan      ;version 1.0
135: fdc      equ 0f8H      ; Base port address FDC

```

(Continued on next page)

EXPERT TOOLKIT

Knowledge-based Inference Program

- **ENTER THE 5th GENERATION** by introducing yourself to the world of "expert" computer systems
- **THE EXPERT TOOLKIT** offers you the information and capability to move to 5th GENERATION programs and/or programming
- **EXPERT - 2 MANUAL** provides tutorial overview of "expert" computer systems and knowledge engineering, including bibliography
- **SIMPLE EXPERT PROJECTS**
 - Stock Market Analyzer
 - Animal Game
 - Digital Fault Analyzer
- **BOOTABLE DISK** with source code for Expert-2 and animal's game
- **5.25" DISK FORMAT** for Apple II series, IBM, Osborne, Kaypro
- EXPERT TOOLKIT *** \$79
 - includes Expert-2 manual, bootable FORTH system disk (FORTH not supported), EXPERT-2 language with source, animal game example with source
- EXPERT - 2 MANUAL ONLY** only \$25
 - * Also available from selected FORTH vendors

Payment by check or money order.

Add \$3.00 First Class mail.

California residents add 6% sales tax.

Dealer inquiries are welcomed.

NIMBLE

Division of Helion, Inc.
Box 445
Brownsville, Ca. 95919
(916) 675-2478

PROGRAMMING IN C ? BE SURE YOU GET ALL THE PHACT's !

base your C programming on
PHACT-dbrm

a multi-keyed (IISAM)
Data Base Record Manager.

PHACT-dbrm

is an easy to use library of C callable functions for manipulation of records in a database, plus high level database manipulation tools.

PHACT-dbrm

supports: data dictionary; 5 datatypes; variable length records; full database security; database locking, data portability; "h" file creation and much much more!

PHACT-dbrm

runs on: all UNIX systems • IDRIS and UNIX look-alikes • MSDOS • CP/M

PHACT-rql (Relational Query Language) **PHACT-rs**
(Report Generator) and **PHACT-rsg** (Relational Screen Generator)
under development.

PHACT-dbrm

is priced between \$250-\$950. (Source available)



To get all the PHACT's call DAVID GRAHAM at
PHACT ASSOCIATES Ltd • 212 • 420-1512
231 EAST 11 STREET • NEW YORK • NY 10003

Circle no. 78 on reader service card.

Circle no. 72 on reader service card.

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```

136: ;
137: cmd      equ      fdc          ; command res.
138: stat     equ      fdc          ; status res.
139: trk      equ      fdc+1        ; track res.
140: sec      equ      fdc+2        ; sector res.
141: data     equ      fdc+3        ; data res.
142: dma      equ      fdc+4        ; Dma controller
143: dsel     equ      fdc+5        ; drive select
144:         endif
145:
146:         if      CCS             ;version 1.1
147: fdc      equ      30H           ; Base port address FDC
148: ;
149: cmd      equ      fdc          ; command res.
150: stat     equ      fdc          ; status res.
151: trk      equ      fdc+1        ; track res.
152: sec      equ      fdc+2        ; sector res.
153: data     equ      fdc+3        ; data res.
154: dma      equ      004CH        ;memory address of pointer to
155: ;                               sector buffer
156: dsel     equ      fdc+4        ; drive select port:
157: ;                               ;output BX for single density
158: ;                               ;       FX for double density
159: ;                               ;       where X = 1 for drive A
160: ;                               ;               2 for drive B
161: ;                               ;               4 for drive C
162: ;                               ;               8 for drive D
163:         endif
164:
165: *****
166: *** "FLOPPY CONTROLLER COMMANDS" ***
167: *****
168:
169: ;       These are common to most systems using the 1793.
170: ;       sdma is a Dyan controller command, ignored in V. 1.1.
171:
172: clear    equ      0d0h          ; clear FDC
173: sdma     equ      87h           ; Start Dma Transfer
174: rsec     equ      80h           ; read sector
175: raddr    equ      0c0h          ; read address
176: seek     equ      18h           ; seek track
177: restore  equ      8            ; home Head(s)
178: sdelay   equ      4            ; seek delay flag
179:
180: *****
181: *** "FLOPPY CONTROLLER VARIABLES" ***
182: *****
183:
184: rate     equ      1            ; step rate
185: retry    equ      1            ; read retries
186: fdelay   equ      11          ; FDC delay
187: ;       ; see "delay" routine - this value is for
188: ;       ; 4 Mhz. Z-80; adjust here or in "delay"
189: ;       ; to suit your clock and processor...
190: indxbyt  equ      2            ; byte in status res. when index hole
191: ;       ; is passing sensor
192: busybyt  equ      1            ; byte in status res. when controller
193: ;       ; is busy
194: drabyt   equ      2            ; byte in status res. when data is
195: ;       ; waiting to be read by the computer
196:
197: *****
198: *** "DDD DISKETTE EQUATES" ***
199: *****
200:
201: ;       Adjust to match your diskette size and format.
202:
203: ttrk     equ      76           ; total number of tracks
204: sectors  equ      26           ; sectors per track

```

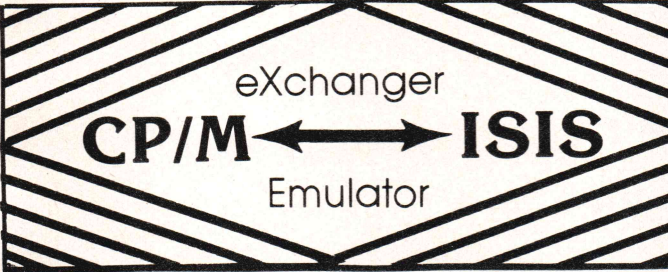


```

205:                                     ; 26 for 8" DDD, 16 for 5.25" DDD
206: lneg      equ      sectors+2          ; last neg. sector +2
207: lpos      equ      sectors+1          ; last pos. sector +2
208: tsec      equ      (sectors/2)-1      ; centering test
209:
210:          if      Dysan
211: ref      equ      798                  ; reference time (single Den)
212:                                     ; (used in "index 7")
213:          endif
214:
215:          if      CCS
216: ref      equ      24eh                  ; experimentally determined...
217:          endif
218:
219:
220:
221: *****
222: *** "MAIN PROGRAM LOOP" ***
223: *****
224:
225:          org      100h
226: ;          =====
227: lxi      sp,stack      ; init stack
228: call     select        ; select drive
229: ;
230: main:
231: lxi      sp,stack      ; reset stack
232: call     cls           ; clear display
233: lxi      h,mel         ; display Menu..
234: call     text
235: call     ci            ; selection?

```

(Continued on next page)



eXchanger
CP/M ↔ ISIS
Emulator

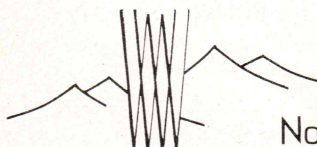
Now move files and programs between your CP/M-80 system and your Intel Series I or II MDS! The ICX package provides complete bidirectional file conversion capability, and even **allows execution** of ISIS-II programs under CP/M using the ISE emulator. The ICX Package is composed of the following two programs:

ICX A Deluxe bidirectional file conversion utility which works with your CP/M system and an 8" floppy drive to provide complete manipulation of an ISIS-II diskette. Takes directories, deletes files, and even initializes a blank disk with the ISIS file structure. Complete C source included. **\$89**

ISE An ISIS-II Emulator which allows ISIS programs to run on any CP/M-80 system. Support for all ISIS-II system and monitor calls makes your CP/M micro look like an MDS! Supports banked memory. Complete MAC source included **\$89**

Complete ICX Package (ICX & ISE) \$175

Supplied on Single Density 8" Disk
CP/M Digital Research, Inc. ISIS-II Intel Corp.



Western Wares
Box C
Norwood CO 81423
(303) 327-4898

RTA for PC

The Ariel RTA is a real time 1/3 octave spectrum analyzer for the IBM Personal Computer. Assembly language routines create an instantaneous display of the frequency spectrum of any audio signal. Also, the analyzer can digitize the signal and store it in the PC's memory for analysis or playback. Call or write for full specifications and applications.

- 31 two pole filters on ISO centers.
- Pink noise source under software control.
- Averaging, weighting and peak hold functions.
- ¼ db resolution from 20 Hz. to 20 KHz.
- 8 bit real time analog input/output system.
- Price: \$649.95 shipping included.

APPLICATIONS

- Aid in room equalization in conjunction with a graphic equalizer.
- Record acoustic response of any enclosure for analysis or comparison.
- Digital storage of raw audio signal for analysis, playback or permanent disk storage.
- Speech research, analysis, synthesis, therapy or recognition.

Ariel —

600 West 116th Street
New York City, N.Y.
10027
(212) 662-7324

Circle no. 122 on reader service card.

Circle no. 5 on reader service card.

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```

236:      cpi      'R'          ; "Radial"
237:      jz       radial
238:      cpi      'A'          ; "Azimuth"
239:      jz       azimuth
240:      cpi      'C'          ; "Centering"
241:      jz       center
242:      cpi      'H'          ; "Hysteresis"
243:      ;      jz       hyster
244:      cpi      'S'          ; "RPM"
245:      jz       rpm
246:      cpi      'I'          ; "Index"
247:      ;      jz       index
248:      cpi      'D'          ; "Drive Select"
249:      jnz      ex
250:      call     select
251:      jmp      main          ; new drive selected - set option
252:
253: ex:      cpi      'E'          ; "Exit to DOS"
254:      jz       exit
255:      mvi      a,bell        ; Not valid
256:      call     co
257:      jmp      main          ; try again...
258:
259:
260: *****
261: *** "EXIT TO DOS" ***
262: *****
263:
264: exit:
265:      call     cls            ; clear display
266:      jmp      wboot
267:
268: *****
269: *** "CONSOLE STATUS" ***
270: *****
271:
272: ;      returns with zero status flag set.
273:
274: cstat:
275:      push     h              ; save all registers
276:      push     d
277:      push     b
278:      mvi      c,11          ; BDOS console status
279:      call     bdos          ; function call added for V1.1
280:      pop      b
281:      pop      d
282:      pop      h
283:      ora      a
284:      ret
285:
286: *****
287: *** "CONSOLE INPUT" ***
288: *****
289: ;
290: ;      Returns character in "A" reg.
291: ci:
292:      push     h              ; Save all registers
293:      push     d
294:      push     b
295:      mvi      c,1          ; BDOS console input
296:      call     bdos          ; function call added for V1.1
297:      pop      b
298:      pop      d
299:      pop      h
300:      cpi      esc           ; Exit?
301:      jz       main          ; YES!
302:      cpi      'a'           ; convert to upper case
303:      rc       ; only...
304:      cpi      'C'

```



```

305:      rnc
306:      ani      ^+1
307:      ret
308:
309:
310: *****
311: *** "CONSOLE OUTPUT" ***
312: *****
313: ;
314: ;      Enter with character in "A" reg.
315:
316: co:
317:      push      PSW
318:      push      h      ; save all registers
319:      push      d
320:      push      b
321:      mov       e,a      ; Pass char in "E" reg for V1.1
322:      mvi       c,6      ; BDOS direct console i/o
323:      call      bdos      ; function call added for V1.1
324:      pop       b
325:      pop       d
326:      pop       h
327:      pop       PSW
328:      ret
329:
330:
331: *****
332: *** "INPUT TRACK LOCATIONS" ***
333: *****
334:
335: ;      Input track locations for Radial
336: ;      Hysteresis, Centering and Index tests.

```

(Continued on next page)

TWO NEW PRODUCTS FOR BDOS C

by A. Skjellum

S-ARGUM COMMAND LINE PROCESSOR

- Bring a uniform command line syntax to all your programs.
- Simplify program construction and maintenance.
- Featured in DDJ, Aug., '82.
- **SUPPORTS:** One letter dash options with optional sub-options and arguments — non-dash arguments.

SOURCE INCLUDED \$50.00

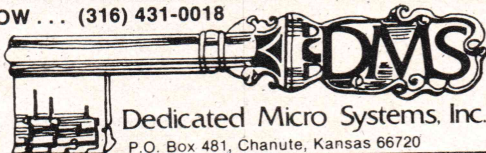
by Robert Ward

MONEY MATH MULTIPLE PRECISION BCD PACKAGE

- Uses BCD internal representation.
- You choose from two types of rounding.
- Configurable exception handling.
- Distributed with 14 digits precision. Easily configured for more or less.
- Excess 64 exponents.

SOURCE INCLUDED \$50.00

CALL NOW ... (316) 431-0018



Dedicated Micro Systems, Inc.
P.O. Box 481, Chanute, Kansas 66720

FOR \$29.95, DISK INSPECTOR MAKES YOU A SHERLOCK HOLMES OF THE COMPUTER!

Ever wonder what happened to that erased or lost file? Did text suddenly disappear and can't be found? Did a bad sector do strange things to your files? Then track them down with *Disk Inspector*! Rated "Excellent" by INFO WORLD, *Disk Inspector* is a utility that pays for itself with the first recovery. Even more, *Disk Inspector* allows you to use the Auto-Load feature of CP/M, blank out bad sectors, create multiple entries, small files, all without any knowledge of programming! Just \$29.95, plus \$2.00 postage and handling, you become the chief inspector. Sold with the usual Overbeek 30 day money-back guarantee.

MAKE ME A CHIEF INSPECTOR! Enclosed find check for \$31.95 or charge my Mastercard # _____ Expires: _____
Visa# _____ Expires: _____

Check format desired:

<input type="checkbox"/> 8" SSDD	<input type="checkbox"/> Osborne Single Density	<input type="checkbox"/> KayPro II
<input type="checkbox"/> Superbrain	<input type="checkbox"/> Osborne Double Density	<input type="checkbox"/> NEC 5"
<input type="checkbox"/> Northstar Advantage	<input type="checkbox"/> Morrow Micro Decision	<input type="checkbox"/> Televideo802
<input type="checkbox"/> Northstar Horizon	<input type="checkbox"/> Xerox 820 Single Density	<input type="checkbox"/> ALTOS 5
<input type="checkbox"/> Apple/Softcard	<input type="checkbox"/> Xerox 820 Double Density	<input type="checkbox"/> Epson

I'm not ready to order now, but send me information about all the affordable programs from Overbeek Enterprises.

Name _____
Address _____
City _____ State _____ Zip _____

OVERBEEK ENTERPRISES, P.O. Box 726D, Elgin, IL 60120
312-697-8420

CP/M is a trademark of Digital Research.

Disk Inspector—another affordable program
from Overbeek Enterprises.

Circle no. 73 on reader service card.

Circle no. 31 on reader service card.

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```

337: ;          "B" contents Upper Limit upon entry.
338:
339: inf$trk:
340:      call    cls          ; clear display...
341:      call    text
342: ;
343: inf$trk1:
344:      call    ci           ; waiting...
345:      cmp     b            ; within limits?
346:      jnc     inf$trk2     ; no...
347:      sui     'A'
348:      jc      inf$trk2     ; try again
349:      lhd     temp        ; table pointer
350:      mvi     d,0
351:      mov     e,a
352:      dad     d            ; add offset
353:      mov     a,m          ; load track
354:      jmp     track       ; seek to track
355: ;
356: inf$trk2:
357:      mvi     a,bell       ; alert
358:      call    co
359:      jmp     inf$trk1     ; try again...
360:
361: *****
362: *** "RADIAL AND HYSTERESIS TRACK LOCATIONS" ***
363: *****
364:
365: rh$tbl

```

MicroScript™ \$99

State of the Art Text Formatter

- extensible generic markup
- multiline headers, footers, and footnotes
- automatic widow and orphan suppression
- automatic section numbering
- automatic bullet, number, and definition lists
- floating and inline figures
- left, center, right, or justify text alignment
- left and right indentation with delay and duration
- bold, underscore, and proportional spacing
- macros and symbols
- multiple input files of unlimited size
- direct printer control
- format preview
- IDS, Qume, Diablo, NEC, C.ITOH, and all TTY
- table of contents, index
- multiple columns
- conditional processing

MicroEd™ \$49

Customizable Full Screen Editor

- full cursor control by character, word, or line
- position to top or bottom of window or file
- scroll by line, half window, or full window
- global or selective find and replace
- delete by character, word, line, or block
- read external files into current file
- copy, move, and write blocks of text
- insert, overlay, or wordwrap text
- all cursor addressable VDTs

Postpaid within U.S. & Canada, outside U.S. add \$5, CA residents add 6 1/2%,
8" SS/SD CPM-86*, 5.25" SS/DD PC-DOS.

VISA

MicroType™

6531 Crown Blvd., Suite 3A, San Jose, CA 95120
(408) 997-5026

Master
Card

* CPM-86 is a trademark of Digital Research, PC-DOS is a trademark of IBM Corporation.

Circle no. 68 on reader service card.

Available now.

The Best of Both Worlds

High-performance

CompuPro Hardware

with high-performance

Cromemco Software.

Cromemco's MC68000-Z80 GROMIX multi-tasking
operating system with drivers to support CompuPro hardware.
(requires Cromemco DPU)

Minimum configuration:

DPU - SystemSupport1 - Disk1 - Interfacer3 or 4 - 192K RAM

CROMIX with drivers to support minimum configuration \$890.

Special drivers only \$295.

The following products can be added to any 8 or 8-16 bit CROMIX system:

SCSI hard disk drivers \$195.

15Mb formatted hard disk subsystem \$2095.
30Mb (two drives) \$2995.

MDrive-H drivers \$195.

CompuPro 512K MDrive-H hardware \$1695.

Other drivers in development... custom inquiries welcome.

Authorized CompuPro Dealer.

PuterParts®

2012 Third Avenue

Seattle, Washington 98121

a Katharos Companies company

Telephone 206-625-9556

OPERATING HOURS

Tuesday Wednesday & Thursday 3 to 6 pm
Friday & Saturday 11 am to 7 pm
Closed Sunday & Monday
Modem access after hours.

"PuterParts" is a registered trademark
of Katharos Companies

Additional trademarks: Z80, Zilog, MC68000, Motorola, CROMIX, DPU, Cromemco,
SystemSupport1, Disk1, Interfacer3, 4, MDrive-H, CompuPro.

Circle no. 83 on reader service card.


```

366:      db      0
367:      db      3
368: ;      Note:  some Dysan spec sheets list track 5 rather
369: ;      than 3 as an index track - my Revision A DDD definitely
370: ;      uses track 3...  LA.
371:      db      38
372:      db      41
373:      db      70
374:      db      73
375:
376: *****
377: *** "INDEX TRACK TABLE" ***
378: *****
379:
380: ix$tbl:
381:      db      0
382:      db      76
383:
384: *****
385: ** "CENTERING TRACKS" ***
386: *****
387:
388: cen$tbl:
389:      db      35
390:      db      44
391:      db      47
392:
393:
394: *****
395: *** "PRINT DECIMAL NUMBERS" ***
396: *****
397: ;

```

(Continued on next page)

DeSmet C \$109

PCDOS+ — CP/M-86++ — MP/M-86++ — CCP/M-86++

• C DEVELOPMENT PACKAGE

C Compiler, Assembler, Linker, and Librarian
Full Screen Editor — Native 2.0 Support

• COMPLETE IMPLEMENTATION

Everything in **K & R** (incl. **STDIO**)
Small case — 64K code, 64K data/stack
All 8088/8087 data types
Intel assembler mnemonics

• OUTSTANDING PRICE/PERFORMANCE

SIEVE Benchmark
135 bytes compiled — 6144 bytes linked
65 sec. compile (fdisk) — 11.5 sec. run (10 iter)

• SUPPORT LIBRARIES

Both 8087 and Software Floating Point
PC Keyboard/Display
Trig/Log functions
Basic Graphics [4th Qtr 83]

• UPDATES

Updates and Bugs reported in newsletter
No time/number limits

+ Trademark IBM ++ Trademark Digital Research

ORDER FORM

Name _____

Address _____

Phone _____

OS: ☐ PCDOS ☐ CP/M-86, MP/M-86, CCP/M-86

DISKS: ☐ 8" (3740) ☐ 5.25" SS (160KB) ☐ 5.25" DS (320KB)

DeSmet C Package \$109 _____

Manual Only (refunded with order) \$20 _____

Update \$20 _____

Sub Total _____

California Residents add 6 % Sales Tax _____

Send To: **C WARE**

Shipping _____

970 W. El Camino Real, Suite 4

Total _____

Sunnyvale, CA 94087
(408) 736-6905

Circle no. 28 on reader service card.

LISP

FOR THE IBM PERSONAL COMPUTER.

THE PREMIER LANGUAGE
OF ARTIFICIAL
INTELLIGENCE FOR
YOUR IBM PC.

DATA TYPES

Lists and Symbols
Unlimited Precision Integers
Floating Point Numbers
Character Strings
Multidimensional Arrays
Files
Machine Language Code

MEMORY MANAGEMENT

Full Memory Space Supported
Dynamic Allocation
Compacting Garbage Collector

FUNCTION TYPES

EXPR/FEXPR/MACRO
Machine Language Primitives
Over 190 Primitive Functions

IO SUPPORT

Multiple Display Windows
Cursor Control
All Function Keys Supported
Read and Splice Macros
Disk Files

POWERFUL ERROR RECOVERY

8087 SUPPORT

COLOR GRAPHICS

LISP LIBRARY

Structured Programming Macros
Editor and Formatter
Package Support
Debugging Functions
.OBJ File Loader

RUNS UNDER PC-DOS 1.1 or 2.0

IQLISP

5 1/4" Diskette
and Manual _____ \$175.00
Manual Only _____ \$ 30.00

 **Integral Quality**

P.O. Box 31970
Seattle, Washington 98103-0070
(206) 527-2918

Washington State residents add sales tax.
VISA and MASTERCARD accepted.
Shipping included for prepaid orders.

Diagnostic Diskette Listing

(Listing continued, text begins on page 40)

```

398: ;      Enter with number to be printed in
399: ;      'DE' res.
400:
401: pnum:
402:      push    h      ; Setup Stack
403:      lxi     h,10    ; Terminator...
404:      push    h
405:      push    h
406: ;
407: pnt1:
408:      call    div16    ; Divide Number By (10)
409:      mov     a,d      ; Check result
410:      ora     e        ; = 0
411:      jz      pnt2
412:      xthl
413:      dcr     l
414:      push    h
415:      lxi     h,10
416:      jmp     pnt1      ; Divide Again
417: ;
418: pnt2:
419:      pop     d        ; Display Values
420:      mov     e,l
421: ;
422: pnt3:
423:      mov     a,e      ; Check for Terminator (10)
424:      cpi     10
425:      pop     d        ; Next digit
426:      rz
427:      adi     '0'      ; Add ascii Bias
428:      call    co
429:      jmp     pnt3
430:
431:
432: *****
433: *** "16 BIT SUBTRACTION" ***
434: *****
435: ;
436: ;      Subtract "DE" from "HL"
437: subde:
438:      mov     a,l
439:      sub     e
440:      mov     l,a
441:      mov     a,h
442:      sbb     d
443:      mov     h,a
444:      ret
445:
446:
447: *****
448: *** "COMPARE "DE" TO "HL" ***
449: *****
450:
451: dehl:
452:      mov     a,l      ; compare "E" to "L"
453:      cmp     e        ; If not zero return
454:      rnz
455:      mov     a,h
456:      cmp     d
457:      ret
458:
459:
460:
461: *****
462: *** "16/24 BIT DIVIDE ROUTINE" ***
463: *****
464:
465: ;      Divide "HL" by "DE"

```



```

466: ;
467:
468: div16:
469: xra      a          ; optional 16 bit divide
470: ;
471: div24:
472: sta      msb          ; Normal 24 bit divide
473: shld     msb1
474: lxi      h,msb2
475: movi     m,24+1
476: lxi      b,0
477: push     b
478: div25:
479: mov      a,e
480: ral
481: mov      e,a
482: mov      a,d
483: ral
484: mov      d,a
485: lda      msb
486: ral
487: sta      msb
488: dcr      m
489: pop      h
490: rz
491: movi     a,0
492: aci      0
493: dad      h
494: mov      b,h
495: add      l
496: lhld     msb1
497: sub      l
498: mov      c,a
499: mov      a,b
500: sbb      h
501: mov      b,a
502: push     b
503: jnc      div26
504: dad      b
505: xthl
506: ;
507: div26:
508: lxi      h,msb2
509: cmc
510: jmp      div25
511:
512:
513:
514: *****
515: *** "CLEAR CONSOLE DISPLAY" ***
516: *****
517:
518: ;      Clear console to foreground spaces.
519:
520: cls:
521:      if      Hazel
522:      movi     a,lead$in      ; lead in
523:      call    co
524:      endif
525:
526:      movi     a,clr$co      ; clear to foreground spaces
527:      jmp      co
528:
529: *****
530: *** "CLEAR DISPLAY LINE" ***
531: *****
532:
533: cline:
534:      if      lineclr
535:      movi     a,0      ; starting column
536:      sta      column

```

(Continued on next page)

FORTH 79 STANDARD
with extensions for the
IBM PC

HS / FORTH

ADVANCED PROGRAMMING
MAKES HS-FORTH THE
FASTEST FULL MEGABYTE
AVAILABLE

COMPARE FOR YOURSELF
WITH THE BYTE
"SIEVE OF ERATOSTHENES"
BENCHMARK jan83

BASIC	2000 sec
other FORTH's	70-140 sec
(most limited to 64K program size)	

HS-FORTH	47 sec
with AUTO-OPT	9 sec
with MICRO-ASM	5 sec
ASSEMBLER	5 sec

AUTO-OPT requires NO
assembly language knowledge.
MICRO-ASM is the simplest of
assemblers to use — and has
the same high level control
structures as FORTH.
Both are included at no extra
charge.

Whether you're a BASIC pro-
grammer that wants the power
of 200 PC's or an ASM hacker
that wants the programming
ease of an interactive inter-
preter—

HS-FORTH IS THE ANSWER!

HS-FORTH	\$220.
with AUTO-OPT and MICRO-ASM	

HARVARD SOFTWARES

PO BOX 339
HARVARD, MA 01451
(617) 456-3021

Circle no. 45 on reader service card.

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```

537:      lxi      h,set$cur      ; set cursor
538:      call     text
539:      endif
540:
541:      if       Hazel AND lineclr
542:      mvi      a,lead$in      ; lead in byte
543:      call     co
544:      endif
545:
546:      if       lineclr
547:      mvi      a,end$ln      ; clear to end of line
548:      call     co
549:      endif
550:
551:      if       NOT lineclr
552:      mvi      a,11          ; starting column
553:      sta      column
554:      lxi      h,set$cur      ; set cursor
555:      call     text
556:      lxi      h,blanks      ;address of a string of 56 blanks...
557:      call     text
558:      endif
559:
560:      ret
561:
562: *****
563: *** "FATAL ERROR MSG" ***
564: *****
565:
566: fatal:
567:      if       lineclr
568:      call     cline          ; clear line
569:      endif
570:
571:      mvi      a,30          ; column position
572:      sta      column
573:      lxi      h,set$cur
574:      call     text
575:      lxi      h,me2          ; Fatal error msg
576:      call     text
577:      ret
578:
579:
580: *****
581: *** "PRINT STRING OF TEXT" ***
582: *****
583:
584: ;      Enter with "HL" pointing to text string,
585: ;      terminate string with Null.
586: ;
587: ;      Direction cursor position is used for displays.
588: ;      The value (-1) at the start of string indicates
589: ;      to this routine
590: ;      that column and row position will follow.
591: ;      This format does not necessarily go out to
592: ;      your terminal. If your terminal needs row sent before column,
593: ;      or a constant bias added (0,0 is not the upper left
594: ;      corner of the screen), you'll have to invent code
595: ;      within this routine to accomplish that.
596: ;
597: ;      This is the internal representation format:
598: ;      db -1,50,10,'NOW IS THE TIME'
599: ;          ^ ^ ^
600: ;          | | | _Row position (add bias if needed)
601: ;          | | | _Column position (add bias if needed)
602: ;          | | | _Flag (send console lead-in sequence.
603:
604: text:
605:      mov      a,m            ; load char.

```



```

606:      ora      a          ; end of string?
607:      rz              ; yes
608:      cpi      (-1) AND 0FFH ; position cursor?
609:                      ; "and offh" added for ASM compatibility
610:      jz       text1     ; yes
611:      inx      h          ; next char.
612:      call     co        ; output
613:      jmp      text
614: ;
615: text1:
616:      inx      h          ; move passed command
617:
618:      if       Hazel
619:      mvi      a,lead$in   ; leading for console
620:      call     co        ; issue.
621:      endif
622:
623:      mvi      a,add$cur   ; address cursor
624:      call     co
625:      mov      a,m        ; column position...
626:      inx      h
627:      call     co
628:      mov      a,m        ; row position
629:      call     co
630:      inx      h
631:      jmp      text
632:
633: *****
634: *** "CENTERING CHECK" ***
635: *****
636:
637: ;

```

(Continued on next page)

SUPER-FAST!

Z80

Relocating Macro
ASSEMBLER
\$169.95

Directly generates COM, HEX, or REL files. Flexible REL format allows external bytes and words with complete math operations on them (Microsoft format optional). Features Zilog mnemonics with nested macros, conditionals and include files. Unique one-pass design generates compact intermediate code which is then processed to resolve forward references, yielding tremendous time savings. Complete listing, symbol table and cross-reference output may be sent to any device. The perfect tool for assembly language programming. Linker included. Manual only — \$30. For Z80 CP/M and TRS-80.

S L R Systems

1622 NORTH MAIN ST. • BUTLER, PA 16001 • (412) 282-0864

Shipping \$3 U.S., others \$7. PA + 6%. Check, MO, VISA, M/C, C.O.D. Most formats available. Dealer and OEM inquiries invited. Z80, CP/M, TRS-80 TM's of Zilog, Digital Research, Tandy Corp.

Circle no. 94 on reader service card.

AT LAST!

A PROFESSIONAL JOURNAL FOR ENGINEERS
SCIENTISTS MATHEMATICIANS & STATISTICIANS
USING MICROCOMPUTERS.

ACCESS

- *numerical analysis
- *math modeling
- *statistical analysis
- *computerized design
- *process simulation
- *report generation

The articles in ACCESS are written by working engineers and scientists who share their knowledge of how to make productive use of microcomputers with you. Your subscription to ACCESS will make your microcomputer more useful in all areas where engineers and scientists use microcomputers. And you'll even find ways to use your computer you hadn't thought of. The articles in ACCESS are written with you in mind and are aimed at helping you turn your microcomputer into the most productive tool possible. Join the other engineers and scientists who make ACCESS their source of information on microcomputer applications. Subscription rates are 6 issues for \$16. (Canada & Mexico \$20. Other \$32). Fill out the coupon below TODAY. Send check, money order, purchase order, or use your VISA or MASTER CARD.

Sign me up. \$16 ☐ enclosed ☐ bill me ☐ bill my company

Charge ☐ VISA ☐ MC # _____

Exp _____

☐ Send sample issue here's \$3

Name & Address _____

City State and ZIP _____

Mail to ACCESS PO Box 12847 Research Triangle Park, NC 27709 Published by LEDS Publishing Co., Inc.

Circle no. 57 on reader service card.

"Extremely Useful program for Microsoft Basic users."

Jerry Pournelle
"The User's Column"
Byte Magazine, April 1983

WE DO WINDOWS!

Spaghetti code is what many experts call a beginner's Basic program which is all tangled up and difficult to follow. The **Active Trace** package will help you learn how to avoid the pitfalls of structureless programs. And if you already have a program which is too confusing to follow, or has an error which is hiding, relax. **Active Trace** doesn't get confused. **Active Trace** pops a window into Basic that lets you see inside your program as you run it. It will lead you through your program letting you know variable values (all variables or just those you specify) as they change. In a form a novice can understand, your program's internal activity is presented on your screen, or printer, or it can be saved on disk.

WE EVEN SAVE WINDOWS

For more advanced programmers, creating a disk file of a program's activity can make debugging virtually automatic. If X=42, and it shouldn't and you can't figure out where you went wrong, you can instruct your word processor to find "X=42" in the activity file created by **Active Trace's Scope**. The line number responsible for your problem, the line in which X became equal to 42, will appear on your screen! You'll also see a display of every other variable and function value at that point in program execution. You can then step backwards through your program's "real time" activity if the problem had its roots elsewhere.

To order or request information
 contact your dealer or
CALL TOLL FREE 800-358-9120 (US)
 In California call 800-862-4948

Active Trace

Complete package includes Scope, XREF
 mapping and documentation

\$79.95
 was
 \$125.00

Scope Separately

Only recommended for those who already
 own professional mapping (XREF) programs

\$49.95

Documentation

Applicable to purchase

\$12.50

Active Trace is available for most MS-DOS and CPM 2.2 systems and supports the special features of Brand specific versions of Microsoft Basic such as **Basica** on the IBM-PC.

AWARECO
 Active Software

P.O. Box 695 Gualala, CA 95445
 (707) 884-4019

Active trace, Active software, AWARECO and Scope are trademarks of A Ware Company—CPM is a trademark of Digital Research—MS-DOS and Microsoft are trademarks of Microsoft Corporation—IBM-PC is a trademark of IBM Corp.

Diagnostic Diskette Listing

(Listing continued, text begins on page 40)

```

638: ;      Centering test requires the FDC to read
639: ;      all the sectors before diskette centering
640: ;      is consider OK.
641:
642: center:
643:     lxi      h,cen$tbl      ; track table
644:     shld     temp
645:     lxi      h,me9          ; tracks with
646:     mvi      b,'D'          ; limit
647:     call     inf$trk        ; seek track
648:     call     cls            ; clear display
649:     lxi      h,me10         ; centering msg.
650:     call     text
651:     lxi      h,me15         ; frame
652:     call     text
653:     lxi      h,me11         ; New track msg
654:     call     text
655: ;
656: center1:
657:     call     rd$trk          ; read centering track
658:     mvi      a,10           ; set row position
659:     sta      row
660:     jc       center3        ; fatal error..
661:
662:     if       lineclr
663:     call     cline          ; clear line
664:     endif
665:
666:     mvi      a,30           ; set cursor...
667:     sta      column         ; column position
668:     lxi      h,set$cur
669:     call     text
670:     lda      diff           ; must = 0
671:     ora      a
672:     lxi      h,me3          ; "RE-CLAMP DISKETTE"
673:     jnz      center2
674:     mov      a,c
675:     cpi      tsec           ; all sec read?
676:     jnz      center2
677:     lxi      h,me4          ; "CENTERING OK"
678: ;
679: center2:
680:     call     text
681:     call     cstat          ; abort?
682:     jz       center1
683:     call     ci
684:     cpi      ' '            ; new track?
685:     jnz      center1        ; no...
686:     jmp      center         ; yes..
687: ;
688: center3:
689:     call     fatal          ; fatal error
690:     jmp      center2
691:
692:
693: *****
694: *** "AZIMUTH ALIGNMENT CHECK" ***
695: *****
696:
697: azimuth:
698:     call     cls            ; clear display...
699:     mvi      a,76           ; seek to azimuth track
700:     call     track          ; move...
701:     lxi      h,me12         ; type of test msg
702:     call     text
703:     lxi      h,me15         ; frame
704:     call     text
705: ;

```

Circle no. 8 on reader service card.

SMALL TALK 80

The promised wonders of object-oriented programming have arrived. After years of intensive research, Xerox (in collaboration with Addison-Wesley) has released the first in a series on Smalltalk - 80.

Smalltalk - 80: The Language and Its Implementation describes the specifications for a Smalltalk - 80 virtual machine. Although not meant as a language tutorial, the text is eminently useful as a reference for the experienced computer programmer.

Please send _____ copies of *Smalltalk - 80: The Language and Its Implementation* @ \$34.95 to:

Name _____
Company _____
Address _____
City _____ State _____ Zip _____
I enclose \$ _____ by
☐ Check ☐ Money order
☐ MC ☐ VISA
Card # _____
Interbank# _____ Exp. Date _____
Signature _____

SHIPPING INCLUDED
(Allow 4-6 weeks for delivery)

**Peterborough
Distribution
Services
PO Box 458
Peterborough, NH 03458
(603) 924-3843**

```

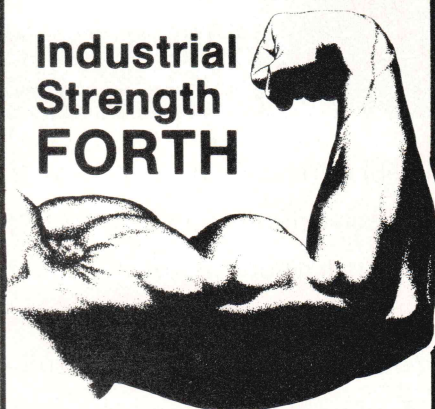
706: azimuth1:
707:     lxi     h,azi$tab      ; set translation table
708:     shld    xpoint        ; pointers
709:     shld    ypoint
710:     call    rd$trk         ; read azimuth track
711:     movi    a,10           ; set row position
712:     sta     row
713:     jc      azimuth3      ; fatal error...
714:
715:     if      lineclr
716:     call    cline          ; clear line
717:     endif
718:
719:     movi    a,20           ; Display negative angle
720:     sta     column
721:     lxi     h,set$cur      ; set cursor
722:     call    text
723:     movi    a,'-'         ; direction of angle
724:     call    co             ; output...
725:     movi    a,' '         ; space
726:     call    co
727:     lhld    ypoint        ; translation
728:     mov     e,m           ; data negative...
729:     movi    d,0
730:     call    pnum           ; display
731:     lxi     h,me5          ; print minutes
732:     call    text
733:     movi    a,45           ; Display Positive angle
734:     sta     column
735:     lxi     h,set$cur
736:     call    text
737:     movi    a,'+'         ; direction of angle
738:     call    co             ; output...
739:     movi    a,' '         ; space
740:     call    co
741:     lhld    xpoint        ; translation
742:     mov     e,m           ; data positive..
743:     movi    d,0
744:     call    pnum           ; display
745:     lxi     h,me5          ; print minutes
746:     call    text
747: ;
748: azimuth2:
749:     call    cstat          ; abort?
750:     jz      azimuth1
751:     call    ci             ; test...
752:     jmp     azimuth1       ; no..
753: ;
754: ; Fatal error Has occurred
755: ;
756: azimuth3:
757:     call    fatal          ; fatal read error
758:     jmp     azimuth2
759:
760:
761: *****
762: *** "RADIAL ALIGNMENT CHECK" ***
763: *****
764:
765:
766: radial:
767:     lxi     h,rh$tbl       ; table pointer
768:     shld    temp
769:     lxi     h,me7          ; track display..
770:     movi    b,'G'         ; limit
771:     call    in$trk        ; select track and seek
772:     call    cls
773:     lxi     h,me8          ; print scale...
774:     call    text
775:     lxi     h,me11         ; Space bar msg
776:     call    text

```

(Continued on next page)

Multiuser/Multitasking
for 8080, Z80, 8086

Industrial Strength FORTH



TaskFORTH™

The First
Professional Quality
Full Feature FORTH
System at a micro price*

LOADS OF TIME SAVING PROFESSIONAL FEATURES:

- ☆ Unlimited number of tasks
- ☆ Multiple thread dictionary, superfast compilation
- ☆ Novice Programmer Protection Package™
- ☆ Diagnostic tools, quick and simple debugging
- ☆ Starting FORTH, FORTH-79, FORTH-83 compatible
- ☆ Screen and serial editor, easy program generation
- ☆ Hierarchical file system with data base management

* Starter package \$250. Full package \$395. Single user and commercial licenses available.

If you are an experienced FORTH programmer, this is the one you have been waiting for! If you are a beginning FORTH programmer, this will get you started right, and quickly too!

**Available on 8 inch disk
under CP/M 2.2 or greater
also
various 5¼" formats
and other operating systems**

**FULLY WARRANTIED,
DOCUMENTED AND
SUPPORTED**



DEALER
INQUIRES
INVITED



Shaw Laboratories, Ltd.
24301 Southland Drive, #216
Hayward, California 94545
(415) 276-5953

Diagnostic Diskette Listing

(Listing continued, text begins on page 40)

```

777: ;
778: radial1:
779: ;
780:         lxi         h,rad$pos      ; radial positive
781:         shld        xpoint         ; cursor table
782: ;
783:         lxi         h,rad$neg      ; radial negative
784:         shld        ypoint         ; cursor table
785: ;
786:         call        rd$trk         ; read radial track
787:         mvi         a,11          ; set row position
788:         sta         row
789:         jc          radial4        ; Fatal error has occurred
790:         call        cline          ; clear line (1)
791:         lda         row
792:         push        PSW
793:         inr         a              ; clear line (2)
794:         sta         row
795:         call        cline
796:         POP         PSW           ; reset line position
797:         sta         row
798:         lhld        ypoint         ; negative
799:         mov         a,m
800:         sta         column         ; position cursor
801:         mov         b,a
802:         lxi         h,set$cur
803:         call        text
804:         mvi         a,'^'         ; set pointer
805:         call        co
806:         lhld        xpoint
807:         mov         a,m           ; positive position
808:         sta         column
809:         mov         c,a           ; save cursor position
810:         lxi         h,set$cur
811:         call        text
812:         mvi         a,'^'
813:         call        co
814:         mov         a,b
815:         sta         column         ; reset column position
816:         lda         row           ; down one line
817:         inr         a
818:         sta         row
819:
820:
821: *****
822: *** "SET POINTERS AND DRAW LINE" ***
823: *****
824:
825: ;         used to draw the graphic display of the
826: ;         radial error.
827:
828:         lxi         h,set$cur      ; position cursor
829:         call        text
830:         mov         a,c
831:         sub         b              ; line length
832:         dcr         a              ; less (1)
833:         mov         b,a
834:         mvi         a,'!'         ; pointer...
835:         call        co
836: ;
837: radial2:
838:         mvi         a,'_'         ; Jointing line
839:         call        co
840:         dcr         b              ; line length -1
841:         jnz        radial2
842:         mvi         a,'!'
843:         call        co
844: ;

```



```

845: radial3:
846:      call    cstat      ; check for abort
847:      jz      radial1    ; read again
848:      call    ci         ; new track?
849:      cpi     ' '
850:      jnz     radial1    ; no...
851:      jmp     radial     ; new track...
852:
853: *****
854: *** "FATAL RADIAL ERROR" ***
855: *****
856:
857: radial4:
858:
859:      if      lineclr
860:      call    cline      ; clear display
861:      lda     row        ; next line
862:      inr     a
863:      sta     row
864:      call    cline
865:      endif
866:
867:      call    fatal      ; fatal read error
868:      jmp     radial3    ; abort?
869:
870:
871: *****
872: *** "CHECK SPINDLE SPEED" ***
873: *****
874: ;
875: ;      All timing is based on 4mhz Z80 CPU
876: ;
877: rpm:
878:      call    cls        ; clear display
879:      lxi     h,me19     ; test performed
880:      call    text
881:      lxi     h,me15     ; frame...
882:      call    text
883:      mvi     a,10       ; set row
884:      sta     row
885: ;
886: rpm1:
887:      call    rpm2       ; compute time in 100us
                        ; increments
888:      shld    temp       ; store count
889:
890:      if      lineclr
891:      call    cline      ; clear line
892:      endif
893:
894:      mvi     a,19       ; set cursor
895:      sta     column
896:      lxi     h,set$cur
897:      call    text
898:      lhld    temp
899:      xch     ; convert to milliseconds
900:      lxi     h,10
901:      call    div16
902:      push    h          ; Save remainder
903:      call    pnum
904:      mvi     a,'.'     ; decimal point
905:      call    co
906:      pop     h
907:      xch
908:      call    pnum       ; fraction
909:      lxi     h,me20     ; milliseconds ms$
910:      call    text
911:      mvi     a,50
912:      sta     column
913:      lxi     h,set$cur
914:      call    text

```

(Continued on next page)

Write more powerful applications with Idris, Whitesmiths' more powerful operating system.

To produce the high quality applications today's business environments require, you need a powerful, sophisticated operating system. Idris, the fastest, most complete UNIX-like system for micros, meets your needs. Idris offers:

- multi-user, multi-tasking capabilities
- networking, even between dissimilar computers
- application portability from Idris to UNIX and UNIX to Idris
- ROM-ability

SEND FOR OUR FREE BOOKLET
"THE PROFIT-BUILDERS' CHECKLIST"

Our new booklet gives you a quick and easy way to check the benefits of Whitesmiths' full line of products.

See for yourself how you can produce more powerful applications, cut your costs, widen your markets, and build your bottom line.

Idris is a trademark of Whitesmiths, Ltd.; UNIX is a trademark of Bell Laboratories.



Whitesmiths, Ltd.

97 Lowell Road
Concord, Mass. 01742
(617) 369-8499

NAME _____
TITLE _____
COMPANY _____
ADDRESS _____
CITY _____
STATE _____ ZIP _____
TELEPHONE _____

The best of UNIX™ and more for CP/M™

CLIP/\$49.95

No risk 15 day money back guarantee*

CLIP runs as a standard CP/M 2.2 program replacing the console command processor with a powerful UNIX-like shell. CLIP, optimized for the Z80, takes only 5.2K of additional memory when running your applications programs.

• Over 50 resident commands

• Editing keys

Single keystroke commands allow you to edit and recall your last 10 commands —just like a word processor! These editing keys may be used within most existing programs, too.

• UNIX-like enhancements

CLIP brings the most powerful UNIX features, namely, I/O redirection and pipes, to CP/M.

• Design your own commands

CLIP is also a powerful "macro" programming language replacing SUBMIT with:

- Conditional command execution
- Argument and switch parsing
- Command tracing
- File I/O (OPEN, GET, PUT, CLOSE)
- Extensive string manipulations

• Built-in calculator

This binary, octal, decimal, and hex calculator can pass its results to programs, macros, or 10 memories.

• Built-in universal text editor

• And much more!

On-line HELP, file searching, user defined prompt, and multiple commands per line.

Software Tools Package \$25

A set of fourteen software tools, inspired by UNIX, complement and enhance CLIP. These tools contain: a sorter, binary file editor, resynchronizing file comparer, pattern matcher, word extractor, and more!

File Encryptor \$25

This program will keep letters, data, programs, or any CP/M file secure.

CLIP, Tools and Encryptor \$99.95

Experience more productivity and control of CP/M today! With our money back guarantee, you have nothing to lose!

Visa/MC/Check/MO/PO accepted
Add \$3.50 shipping, AZ: 7% tax

Formats: 8"SSSD/Osborne/Kaypro/Apple/
Heathkit/Northstar/VT180+Rainbow/Xerox 820/
Micro Decision/Superbrain/NEC-8001

*10% restocking fee

Thoughtware, Inc.

800-821-6010 602-327-4305

Orders Technical

P.O. Box 41436
Tucson, AZ 85717

Diagnostic Diskette Listing

(Listing continued, text begins on page 40)

```

915:      lhd      temp      ; convert to RPM
916:      movi     a,9        ; divide By 600,000
917:      lxi      d,27c0h
918:      call     div24
919:      push     h          ; remainder
920:      call     fnum
921:      movi     a,'.'      ; fraction
922:      call     co
923:      pop      h
924:      xch      h
925:      call     fnum
926:      lxi      h,me21     ; rpm msg
927:      call     text
928:      call     cstat      ; abort?
929:      jz       rpm1       ; No..
930:      call     ci         ; Esc key?
931:      jmp      rpm1       ; No..
932:
933:
934: *****
935: *** "INDEX TO INDEX TIMING" ***
936: *****
937:
938: ;
939: ;      With 4 Mhz. Z-80,
940: ;      returns time in 100 Us increments to caller
941: ;      in "HL" regss.
942:
943: rpm2:
944:      call     clpend     ; Clear pending commands
945:      lxi      h,0        ; Clear Counter
946: rpm3:
947:      in       stat       ; Loop Until Index
948:      ani      2          ; Mask All But Index Bit
949:      jnz      rpm3       ; No Index
950: rpm4:
951:      in       stat       ; if Index...
952:      ani      2          ; Wait for No Index
953:      jz       rpm4       ; In Index Hole
954:
955: ;      Add index hole to count
956: rpm5:
957:      call     rpm7       ; 4.25
958:      in       stat       ; 2.75
959:      ani      2          ; 1.75
960:      jnz      rpm5       ; 2.50
961:                          ; Total = 11.25
962: ;      Count until next Index
963:
964: rpm6:
965:      call     rpm7       ; 4.25
966:      in       stat       ; 2.75
967:      ani      2          ; 1.75
968:      jz       rpm6       ; 2.50
969:      ret              ; total = 11.25
970:
971: *****
972: *** "RPM TIMING LOOP" ***
973: *****
974: ;
975: ;      The loop time must be adjusted for 100us
976: ;      increments at your clock speed with your processor.
977: ;
978: ;      note: "RPM5 or RPM6" loop time must be counted.
979: ;
980:
981: rpm7:
982:      inx      h          ; 1.50

```



```

983:      mvi      a,22      ; 1.75
984: ;
985: rpm8:
986:      dcr      a          ; 1.00 (22 * 1.00) = 22 Usec
987:      jnz      rpm8      ; 2.50 (22 * 2.50) = 55 Usec
988:      inx      d          ; 1.50
989:      inx      d          ; 1.50
990:      inx      d          ; 1.50
991:      inx      d          ; 1.50
992:      ret                ; 2.50
993:      ; total = 88.75
994: ;
995: ;
996: *****
997: *** "READ DDD TRACK" ***
998: *****
999: ;
1000: ;      Att: Main read routine for program.
1001: ;      =====
1002: ;
1003: ;      Reads the Positive and Negative offset
1004: ;      sectors and returns pointers to the
1005: ;      translation tables set by caller.
1006: ;
1007: ;
1008: rd$trk:
1009:      MVI      A,7        ;Upper case code adds a simple display of
1010: ;                        ; last sector read (+ & -), giving
1011: ;                        ; ? and @ if first sectors fail
1012: ;                        ; and A thru Z representing the 26
1013: ;                        ; possible sectors on an 8" disk
1014: ;                        ; or A thru P for a 5.25" disk
1015:      STA      ROW
1016:      MVI      A,39
1017:      STA      COLUMN
1018:      LXI      H,SET$CUR
1019:      CALL     TEXT
1020:      mvi      c,lpas      ; last sector
1021:      mvi      a,1          ; beginning sector positive
1022:      call     rd$sec      ; read positive offsets
1023:      ADI      3eH          ; translate sector # to alpha code */
1024:      CALL     CO          ; display on console */
1025:      SUI      3eH          ; return to sector # */
1026:      cpi      1            ; unable to read
1027:      jz       rd$trk2      ; first sec?
1028:      sta      diff        ; store last sector tested
1029:      mvi      c,lnes
1030:      mvi      a,2          ; beginning sector negative
1031:      call     rd$sec      ; read negative offsets
1032:      ADI      3eH
1033:      CALL     CO
1034:      SUI      3eH
1035:      cpi      2            ; fatal error?
1036:      jz       rd$trk2      ; set error flag
1037: ;
1038: ;      Positive sector Translation
1039: ;
1040:      lda      diff
1041:      inr      a            ; adj. sector Num.
1042:      sui      4
1043:      rar                ; divide by two
1044:      lhld     xpoint      ; translate Table pointer
1045:      mvi      d,0          ; setup for Offset
1046:      mov      e,a
1047:      dad      d            ; add offset
1048:      shld     xpoint      ; set pointer
1049: ;
1050: ;      Negative Sector Translation
1051: ;
1052:      in       sec          ; last sector
1053:      sui      4

```

(Continued on next page)

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```

1054:      rar                      ; divide by two
1055:      mov      c,a
1056:      sub      e                      ; diff. for Hysteresis
1057:      sta      diff
1058:      mov      e,c                      ; offset
1059:      lhld     ypoint                ; translation table pointer
1060:      dad      d                      ; offset in table
1061:      shld     ypoint                ; set pointer
1062:      ora      a                      ; clear carry
1063:      ret
1064:
1065: ;      Set fatal error Flag, caller processes
1066: ;      the error.
1067:
1068: rd#trk2:
1069:      stc                      ; carry = Fatal error
1070:      ret
1071:
1072:
1073:
1074: *****
1075: *** "READ SECTORS BY INCREMENTS OF TWO" ***
1076: *****
1077:
1078: ;
1079: ;      Enter with first sector to be read in 'A' reg.
1080: ;      Returns to caller on Error or Last sector.
1081: ;      note:
1082: ;          Z flag set if good read else Nz

```

DISK DRIVES

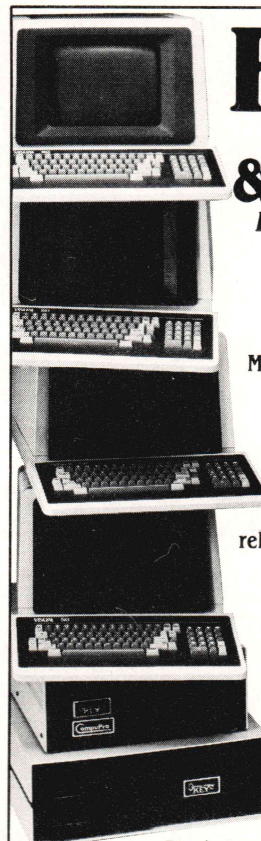
NEW — Full O.E.M. Warranties too!

TANDON

Model Number	Type	tpi	Height	US\$ Price
TM-100-1	SSDD	48	full	177
TM-100-2	DSDD	48	full	220
TM-101-4	DSDD	96	full	270
TM-50-1	SSDD	48	half	160
TM-50-2	DSDD	48	half	200
TM-55-4	DSDD	96	half	250
All Manuals				10
TM-848E-2	8"DSDD	48	half	380
TM-502	12.8mb	Winchester		1000
TM-503	19.8mb	Winchester		1100
TM-703	30.1mb	Winchester		1425
300-1200 Baud RS-232 MODEM				250
Direct connect std. Auto Ans/Originate.				
~Optional Autodial/Smart Pkg.				50

ADLER COMPUTER SYSTEMS COMPANY,
125 Chandler st., Buffalo, N.Y. 14207

Phone [716] 876-1600 | NYS Residents
add 7% s/s.tax



POWER & PRICE

Introducing the KEY DP220
from Key Micro Systems

A complete 4-user (expandable to 8-user) system with dual processor, dual half-height 8" drives and 20 Mbytes. With it you can enjoy the best of an S-100 computer and run any mix of 8 and 16 bit programs simultaneously. Our use of the highest quality peripherals combined with the power of CompuPro boards delivers the reliability you need — at the right price. For established systems, peripheral support, or custom configurations, we deliver the power you want at a powerfully affordable price.

Key Micro Systems, Inc. is an
authorized CompuPro
Systems Center



MICRO SYSTEMS INC.

Terminals optional.

1606 Nooseneck Hill Rd., PO Box 715, Coventry, RI 02816 • 401/828-7270
822 Boylston St., Suite 201, Chestnut Hill, MA 02167 • 617/738-7305


```

1083: rd$sec:
1084:      call    read          ; read sector
1085:      in      sec           ; current sector
1086:      rnz
1087:      adi     2
1088:      out     sec
1089:      cmp     c              ; last sec?
1090:      rz
1091:      jmp     rd$sec         ; yes...
1092:
1093: *****
1094: ***  "THE FOLLOWING ARE HARDWARE DEPENDENT ROUTINES" ***
1095: *****
1096: ;
1097: ;
1098: *****
1099: ***  "DRIVE SELECT" ***
1100: *****
1101: ;
1102: ;      The drive select routine must be expanded
1103: ;      for selected systems.
1104:
1105: select:
1106:
1107:      if      Dysan
1108:      mvi     a,21h          ; select bits
1109:      out     dsel          ; select drive...
1110:      endif
1111:
1112:
1113:      if      CCS           ;The upper case code performs drive
1114:                        ; selection on the CCS 2422 controller
1115:                        ; for single side double density...
1116:      CALL    CLS           ; clear console */

```

(Continued on next page)

Now You Can Afford Another 64K...

**Especially when it's less than
a half cent per bit!**

Specifications:

- Fully Static Operation
- Supports S-100 IEEE-696 Standards
- Uses Popular 2716 Pinout Type Static RAM's
- Board Access Time Under 200nS
- 150nS RAMS Standard
- No Wait States Needed at 6.000MHz
- High Quality FR-4 Type PC Board
- Switch Selectable Phantom Line
- All Data, Status and Address Lines Fully Buffered
- Gold Plated Contact Fingers for Low Contact Resistance and Long Life
- Switch Selectable Extended Address Lines For Up To 16 M-bytes
- Extreme Low Power Dissipation (<500mA Typical)
- Top 8K May Be Switched Disabled and/or Interchangeable with 2716 Type EPROM's

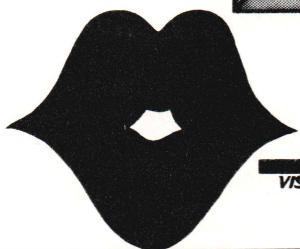
COEX 64K S-100 CMOS

STATIC RAM BOARD

\$299⁰⁰

only

Assembled & Tested



VISA

"Have You Kissed Your Computer Lately?"

Components Express, Inc.

1380 E. Edinger • Santa Ana, Calif. 92705 • 714/558-3972

Terms of Sale: Cash, Checks, Credit Cards, M.O., C.O.D. Calif. residents add 6% sales tax.



Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```

1117:      LXI      H,SELMSG
1118:      CALL     TEXT      ; print drive selection request */
1119:      CALL     CI        ; set response */
1120:      CPI      'A'      ; accept A,B,C,D, as responses */
1121:      JZ       SELECT1
1122:      JC       SELECT
1123:      CPI      'B'
1124:      JZ       SELECT1
1125:      CPI      'C'
1126:      JZ       SELECT3
1127:      CPI      'D'
1128:      JZ       SELECT4
1129:      JMP      SELECT    ; repeat if invalid choice */
1130:
1131:
1132: SELECT4: ADI      3      ; make low nybble an 8 */
1133: SELECT3: INR      A      ; make a 4 */
1134: SELECT1: SUI      'a'    ; make B into 2 or A into 1 */
1135:      ORI      0F0H      ; add high nybble: */
1136:      ; F for double density, B for single
1137:      STA      SELPT     ; store select bits for future use... */
1138:      OUT      DSEL
1139:      MVI      A,40H     ; select side 0... 00 selects side 1... */
1140:      OUT      04        ; secondary control port */
1141:      endif
1142:
1143:      jmp      home      ; home drive
1144:
1145: *****
1146: *** "SEEK HEAD(S) TO TRACK" ***
1147: *****
1148: ;
1149: ;      Enter with new track location in
1150: ;      'A' register.
1151: ;
1152: track:
1153:      out      data      ; new track
1154:      call     clrpend   ; clear FDC
1155:      mvi      a,seek+rate ; seek command + step rate
1156: ;
1157: track1:
1158:      out      cmd        ; issue...
1159: ;
1160:      if      Dysan
1161:      call     delay      ; delay for FDC
1162:      endif
1163: ;
1164:      if      CCS         ; This controller seems to need more
1165:      ; delay - I can't explain why... LA.
1166:      call     delayB     ; delay for FDC
1167:      endif
1168: ;
1169: ;
1170: track2:
1171:      in       stat       ; busy ?
1172:      rrc      ; test
1173:      jc       track2     ; still busy...
1174:      mvi      a,sdelay   ; set seek delay bit
1175:      sta      dflag
1176:      in       trk        ; set current track
1177:      sta      ctrk
1178:      ret
1179: ;
1180:      if      CCS
1181: DELAYB: PUSH     B      ; extra delay for CCS controller... */
1182:      LXI      B,2000H
1183: ;
1184: DELAYC: DCX      B
1185:      MOV      A,C

```



```

1186:      ORA      B
1187:      JNZ      DELAYC
1188:      POP      B
1189:      RET
1190:      endif
1191:
1192:
1193: *****
1194: *** "RESTORE HEAD(S) TO TRACK ZERO" ***
1195: *****
1196:
1197: home:
1198:      call      clPend      ; clear FDC
1199:      mvi       a,restore+rate ; restore command + step rate
1200:      jmp       track1      ; issue command...
1201:
1202: *****
1203: *** "READ SECTOR ROUTINE" ***
1204: *****
1205:
1206: ;      Enter this routine with the sector to
1207: ;      be read in the 'A' register.
1208: ;
1209: ;      This routine will keep trying to read
1210: ;      the sector until retry count equals zero.
1211: ;
1212: ;      The 1793 controller has a built-in automatic
1213: ;      four retries to read ID, so the value set in
1214: ;      the equate (RETRY) is equal to [ 4 * retry + 1].
1215: ;
1216: read:
1217:      out       sec          ; set sector

```

(Continued on next page)

FORTH-79

Ver. 2 For your APPLE II/II+

The complete professional software system, that meets ALL provisions of the FORTH-79 Standard (adopted Oct. 1980). Compare the many advanced features of FORTH-79 with the FORTH you are now using, or plan to buy!

FEATURES	OURS	OTHERS
79-Standard system gives source portability.	YES	_____
Professionally written tutorial & user manual	200 PG.	_____
Screen editor with user-definable controls.	YES	_____
Macro-assembler with local labels.	YES	_____
Virtual memory.	YES	_____
Both 13 & 16-sector format.	YES	_____
Multiple disk drives.	YES	_____
Double-number Standard & String extensions.	YES	_____
Upper/lower case keyboard input.	YES	_____
LO-Res graphics.	YES	_____
80 column display capability	YES	_____
Z-80 CP/M Ver. 2.x & Northstar also available	YES	_____
Affordable!	\$99.95	_____
Low cost enhancement option:		
Hi-Res turtle-graphics.	YES	_____
Floating-point mathematics.	YES	_____
Powerful package with own manual.		
50 functions in all,		
AM9511 compatible.		
FORTH-79 V.2 (requires 48K & 1 disk drive)		\$ 99.95
ENHANCEMENT PACKAGE FOR V.2		
Floating point & Hi-Res turtle-graphics		\$ 49.95
COMBINATION PACKAGE		\$139.95
(CA res. add 6% tax; COD accepted)		

MicroMotion

12077 Wilshire Blvd. # 506
L.A., CA 90025 (213) 821-4340
Specify APPLE, CP/M or Northstar
Dealer inquiries invited.



FORTH-79

Version 2 For Z-80, CP/M (1.4 & 2.x),
& NorthStar DOS Users

The complete professional software system, that meets ALL provisions of the FORTH-79 Standard (adopted Oct. 1980). Compare the many advanced features of FORTH-79 with the FORTH you are now using, or plan to buy!

FEATURES	OURS	OTHERS
79-Standard system gives source portability.	YES	_____
Professionally written tutorial & user manual.	200 PG.	_____
Screen editor with user-definable controls.	YES	_____
Macro-assembler with local labels.	YES	_____
Virtual memory.	YES	_____
BDOS, BIOS & console control functions (CP/M).	YES	_____
FORTH screen files use standard resident file format.	YES	_____
Double-number Standard & String extensions.	YES	_____
Upper/lower case keyboard input.	YES	_____
APPLE II/II+ version also available.	YES	_____
Affordable!	\$99.95	_____
Low cost enhancement options;		
Floating-point mathematics	YES	_____
Tutorial reference manual		
50 functions (AM9511 compatible format)		
Hi-Res turtle-graphics (NoStar Adv. only)	YES	_____
FORTH-79 V.2		\$99.95
ENHANCEMENT PACKAGE FOR V.2:		
Floating point		\$ 49.95
COMBINATION PACKAGE (Base & Floating point)		\$139.95
(advantage users add \$49.95 for Hi-Res)		
(CA. res. add 6% tax; COD & dealer inquiries welcome)		

MicroMotion

12077 Wilshire Blvd. # 506
L.A., CA 90025 (213) 821-4340
Specify APPLE, CP/M or Northstar
Dealer inquiries invited.



Circle no. 66 on reader service card.

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```

1218:      mvi      e, retry      ; to read
1219: ;
1220: read1:
1221:      call      clpend        ; clear FDC
1222:      call      rrs#dma       ; program "DMA" controller
1223:
1224:      if      CCS
1225:      LHLD     DMA            ; set pointer to memory buffer */
1226:      MVI      B, 40H         ; initialize loop counter for I/O */
1227:                                ; transfers: 40 for 8" DD, 20 for 8" SD
1228:      endif
1229:
1230:      lda      dflag         ; seek delay flag
1231:      ori      rsec          ; read sector command
1232:      out      cmd            ; issue..
1233:
1234:      if      Dysan
1235:      mvi      a, sdma        ; start transfer
1236:      out      dma
1237:      endif
1238:
1239:      if      CCS
1240: BREAD: IN      DATA          ; code to perform I/O transfer of */
1241:      MOV      M, A           ; data to memory */
1242:      INX      H
1243:      IN      DATA
1244:      MOV      M, A
1245:      INX      H
1246:      IN      DATA
1247:      MOV      M, A
1248:      INX      H
1249:      IN      DATA
1250:      MOV      M, A
1251:      INX      H
1252:      DCR      B              ; decrement loop counter... */
1253:      JNZ      BREAD
1254:      endif
1255:
1256:      call      comp          ; transfer completed?
1257: ;
1258: read2:
1259:      xra      a              ; clear seek delay
1260:      sta      dflag
1261:      in      stat            ; good read?
1262:      ora      a
1263:      rz              ; yes...
1264:      dcr      e              ; count off retries
1265:      Jnz      read1          ; try again
1266:      in      stat            ; return status
1267:      ora      a
1268:      ret
1269:
1270:
1271: *****
1272: *** "WAIT FOR END OF DMA TRANSFER" ***
1273: *****
1274:
1275: comp:
1276:      call      delay          ; before each check
1277:      in      stat            ; Completed Transfer?
1278:      rrc
1279:      rnc              ; YES!
1280:      jmp      comp
1281:
1282:
1283: *****
1284: *** "PROGRAM DISK CONTROLLER" ***
1285: *****

```



```

1286:
1287: prg$dma:
1288:
1289:     if      Dysan
1290:     mov     b,15      ; Bytes Count
1291:     lxi     h,cmd$tbl ; Command Table
1292: ;
1293: prg$dma1:
1294:     mov     a,m        ; load
1295:     out     dma        ; write to controller
1296:     inx     h
1297:     dcr     b          ; -1
1298:     jnz     prg$dma1
1299:     endif
1300:
1301:     if      CCS
1302:     LXI     H,SECBUF   ; store address of memory buffer */
1303:     SHLD    DMA        ; in pointer for disk controller */
1304:     LDA     SELPT      ; set drive select bits */
1305:     OUT     DSEL       ; sets autowait for 2422 controller... */
1306:                     ; to ensure synchronization of I/O transfer
1307:     endif
1308:
1309:     ret
1310:
1311: *****
1312: *** "COMMAND TABLE DMA CONTROLLER" ***
1313: *****
1314:
1315: ;      This data is used to program Zilog's
1316: ;      z80 DMA controller.

```

(Continued on next page)

BDS C

The *fastest* CP/M-80 C
compiler available today

Version 1.5 contains some nifty improvements:

The unscrambled, *comprehensive* new User's Guide comes complete with tutorials, hints, error message explanations and an index.

The CDB symbolic debugger is a valuable new tool, written in C and included in *source form*. Debug with it, and *learn* from it.

Hard disk users: You can finally organize your file directories sensibly. During compilation, take advantage of the new path searching ability for all compiler/linker system files. And at run-time, the enhanced file I/O mechanism recognizes user numbers as part of simple filenames, so you can manipulate files located *anywhere* on your system.

BDS C's powerful original features include dynamic overlays, full library and run-time package source code (to allow customized run-time environments, such as for execution in ROM), plenty of both utilitarian and recreational sample programs, *and speed*. BDS C takes less time to compile and link programs than any other C compiler around. And the execution speed of that compiled code is typically lightning fast, as the Sieve of Eratosthenes benchmark illustrates. (See the January 1983 BYTE, pg. 303).

BD Software 8" SSD format, \$150
P.O. Box 9 Free shipping on pre-paid orders
Brighton, MA 02135 Call or write for availability on
(617) 782-0836 other disk formats

Circle no. 9 on reader service card.

Changing Your Address?

Staple your label here.

To change your address, attach your address label from the cover of the magazine to this coupon and indicate your new address below.

Name _____

Address _____

Address _____

Apt. # _____

City _____

State _____

Zip _____

Mail to: DDJ, P.O. Box E, Menlo Park, CA 94025

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```
1317:
1318: cmd$tbl:
1319:
1320:     if      Dysan
1321:     db      0c3h          ; Re-Set DMA Controller
1322:     db      8bh          ; Clear Block Counter
1323:     db      79h          ; Recieve Block
1324:     dw      secbuf        ; Address To Store Block
1325:     dw      256          ; Block Size
1326:     db      14h          ; Define Port (B) Address
1327:     db      28h          ; Define Port (A) Address
1328:     db      85h
1329:     db      data          ; Data Res FDC
1330:     db      8ah          ; Set DMA Controller Active HIGH
1331:     db      0cfh
1332:     db      3            ; Data ----> Memory
1333:     db      0cfh
1334:     endif
1335:
1336:
1337: *****
1338: *** "CLEAR PENDING COMMANDS FDC" ***
1339: *****
1340: ;
1341: ;     Clears the floppy controller of any
1342: ;     pending commands.
1343:
1344: clpend:
1345:     mvi     a,clear        ; clear FDC command
1346:     out     cmd            ; issue...
1347:
1348: *****
1349: *** "FDC DELAY TO PROCESS COMMANDS" ***
1350: *****
1351: ;
1352: ;     This delay must be adjusted for different
1353: ;     clock speeds. 50 usec delay loop.
1354: ;
1355: ;     Note:
1356: ;         If altered, Adjust Index Timing Routine.
1357:
1358: delay:          ; 4.25
1359:     mvi     a,fdelay      ; 1.75
1360: delay1:
1361:     dcr     a              ; 11.00
1362:     jnz     delay1        ; 27.50
1363:     nop                    ; 1.00
1364:     nop                    ; 1.00
1365:     nop                    ; 1.00
1366:     ret                ; 2.50
1367:
1368: *****
1369: *** "RADIAL TRANSLATE TABLE" ***
1370: *****
1371:
1372: ;     Cursor Positioning table for radial
1373: ;     alignment test.
1374:
1375: rad$neg:
1376:     db      38            ; 1 millinch
1377:     db      36            ; 2 "
1378:     db      34            ; 3 "
1379:     db      32            ; 4 "
1380:     db      30            ; 5 "
1381:     db      28            ; 6 "
1382:     db      26            ; 7 "
1383:     db      24            ; 8 "
1384:     db      22            ; 9 "
```

(Continued on page 76)

**GGM-FORTH™ has HELP*
for Z80† using CP/M †**

GGM-FORTH, a software system for real-time measurement and control, runs on any Z80 computer under CP/M, using an extended fig-FORTH vocabulary.

GGM-FORTH includes:

- Up to four open CP/M files, in any combination of direct-access and sequential-access, fully compatible with all CP/M utilities
- Char. in/out uses CP/M console, lister, file, or auxiliary port

PLUS

- On-Line HELP* facility, compliments our complete manual
- Separate HELP* files for multiple, user-extendable vocabularies
- Full - Screen editor uses HELP* for better programmer productivity

Complete system and manual: \$150.

GGM SYSTEMS, INC. (617) 662-0550
135 Summer Ave. Reading, MA 01867

†Z80 is a trademark of Zilog, Inc.

†CP/M is a trademark of Digital Research, Inc.

**PROMPT DELIVERY! AND
TRY TO BEAT THESE IC PRICES:**

DYNAMIC RAM

64K	200 ns	\$ 5.65
64K	150 ns	5.95
64K	120 ns	6.27
16K	200 ns	1.56

EPROM

27128	300 ns	\$19.20
2764	250 ns	5.99
2732	450 ns	4.25
2716	450 ns	3.60
2532	450 ns	4.75

STATIC RAM

6264P-15	150 ns	\$38.00
6116P-3	150 ns	5.62
6117P-3	150 ns	4.20

Prices 10/18/83

MasterCard/VISA or UPS CASH COD

Factory New, Prime Parts

µP∞

MICROPROCESSORS UNLIMITED

24,000 South Peoria Ave.

(918) 267-4961

BEGGS, OK. 74421

Prices subject to change. Please expect higher prices on some parts due to world wide shortages. Call for volume prices. Subject to available quantities. Shipping & Insurance extra. Cash discount prices shown. Federal Express Standard Air @ \$5.99! Orders received by 5 PM CST can be delivered to you by the next morning.

Circle no. 42 on reader service card.

Circle no. 67 on reader service card.

**NEW
IBM-PC
DEBUGGER**

*New full screen
debugging concept.
With unique multi-window
multi-level split-screen features:*

- Simple/single keystroke commands
- Full-screen disassemblies may be scrolled through
- Continuous monitoring of selected memory
- Hundreds of tagged breakpoints supported
- Saves and restores user's graphic display when breakpoint hit
- Many other inspired features

Your bugs were never so obvious!

Requires MS-DOS & 96K RAM (min.).

**And now...
right before your very eyes...**

CodeSmith™-86

The screenshot displays the CodeSmith-86 debugger interface. At the top, it shows the file 'fabcode.exe' and the program 'CodeSmith-86'. Below this, there are several windows: a register window showing AX=8086, SP=8087, SS=1983, BX=0000, BP=0000, CS=1984, CX=0000, SI=0000, ES=1985, DX=0000, DI=0000, and IP=0002; an assembly window showing instructions like PUSH BP, PUSH SI, PUSH ES, MOV BP, SP, SUB SP, 40H, MOV FDI(BP), BX, MOV F4H(BP), DX, LEA SI, 1E(BP), REPZ MOVSB, AND AL, 0011100B, and DB 89; and a memory dump window showing the contents of memory starting at address 03C4:0050. The interface is designed for multi-window, multi-level split-screen debugging.

Designed to get your program working—FAST.

VISUAL AGE

642 N. Larchmont Blvd., Los Angeles, CA 90004 • (213) 464-8141

CodeSmith is a registered trademark of International Arrangements, Inc.
Microsoft and MS are registered trademarks of Microsoft Corp. • IBM is a registered trademark of International Business Machines Corp.

INTRODUCTORY OFFER ONLY
California residents
add 6.5% sales tax
\$145.00

Circle no. 120 on reader service card.

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

[illegible]

(Continued on page 78)

Uniforth

One of the finest implementations of the FORTH language. Field tested and reliable, **UNIFORTH** is available for the DEC Rainbow/Professional, Osborne, KayPro, and IBM PC as well as most systems with 8" disks and the following processors:

8080	PDP-11
Z80	68000
8086/8	16032

As a task, **UNIFORTH** is compatible with and supports all features and file types of the CP/M, CDOS, MS-DOS and DEC operating systems. As an operating system, **UNIFORTH** will function "stand-alone" on most commercial microcomputers.

The FORTH-79 Standard language has been extended with over 500 new words that provide full-screen and line-oriented editors, array and string handling, enhanced disk and terminal I/O, and an excellent assembler. Detailed reference manuals supply complete documentation for programming and system operation, in an easy-to-understand, conversational style using numerous examples.

Optional features include an excellent floating-point package with all transcendental functions (logs, tangents, etc.), the MetaFORTH cross-compiler, printer plotting and CP/M file transfer utilities, astronomical and amateur radio applications, word processing, etcetera.

Compare these features with any other FORTH on the market:

- Speed and efficiency
- Variety of options
- Ease of use
- Documentation quality

You'll find **UNIFORTH** is superior.

Prices start at \$35. Call or write for our free brochure.

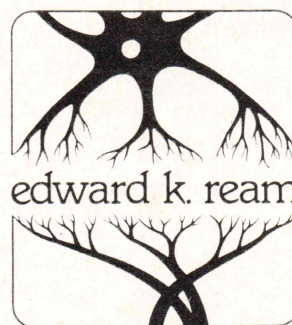
Unified Software Systems

P.O. Box 2644, New Carrollton, MD 20784, (301) 552-9590

CP/M Digital Research, CDOS Cromenco, DEC PDP Digital Equipment Corporation, MSDOS Microsoft, IBM PC IBM, Z80 Zilog

Circle no. 114 on reader service card.

RED



is a

FULL SCREEN EDITOR

with

FULL SOURCE CODE in C

for

CP/M 68K or CP/M 80

- RED is a powerful yet simple text editor for both programmers and writers.
- RED features 17 commands including block move, block copy, search and substitute.
- RED is the only text editor supplied with full source code for the BDS C, Aztec CII and Digital Research C compilers.
- RED is the only text editor that you can change to suit your needs and tastes.
- RED is the only text editor that will not become obsolete when you change systems.
- RED supports all features of your terminal. You tailor RED to your terminal with an easy-to-use configuration program.
- RED handles files as large as your disk.

Price:\$50.

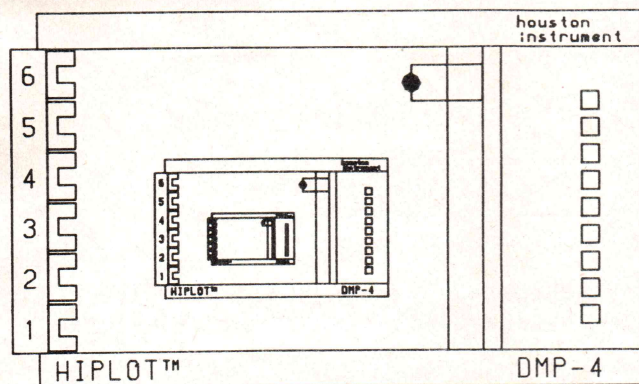
For more information, call (608) 231-2952

To order, write:

Edward K. Ream
1850 Summit Avenue
Madison, Wisconsin 53705

Dealer inquiries invited.

QCAL™ DEVICE-INDEPENDENT CP/M GRAPHICS

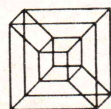


QCAL(tm) emulates the Calcomp(tm) Basic Subroutine Package. The QCAL user (with Microsoft(tm) FORTRAN) may employ the industry standard calls (PLOT, AXIS, SYMBOL, etc.) but utilize many available graphic output devices. A generation of prior graphics application software becomes accessible under CP/M(tm), and new programs using QCAL handle graphics in a time-proven, standardized, and transportable manner. Metric capability is built-in.

QCAL includes (at \$295 on 8" SD diskette) user's manual, sample programs, relative object code for the emulated calls, source for fonts (7 alphabets for USA and Western Europe), and source for one sample graphics device driver. Available choices are HILOT(tm), Watanabe(tm), and NEC Spinwriter(tm). Custom fonts and drivers are easily created using supplied documentation.

QCAL was featured in the US exhibit at "Europe Software 82", The Netherlands. It is now in use around the world.

QCAL (tm) Tesseraet Associates; Calcomp (tm) California Computer Products, Inc.; Microsoft (tm) Microsoft Corp.; CP/M (tm) Digital Research, Inc.; HILOT (tm) Houston Instrument division of Bausch & Lomb; Watanabe (tm) Watanabe Instrument Corp.; Spinwriter (tm) NEC Information Systems, Inc.



TESSERAET ASSOCIATES
STINSON LAKE ROAD
RUMNEY, NH 03266
(603) 786-9561. (617) 964-6740



Circle no. 108 on reader service card.

Diagnostic Diskette Listing (Listing continued, text begins on page 40)

```

1453:      db      (-1) AND 0FFH,25,5,'<>      <>'
1454:      db      (-1) AND 0FFH,25,6,'<>      A = (0)      D = (41) <>'
1455:      db      (-1) AND 0FFH,25,7,'<>      <>'
1456:      db      (-1) AND 0FFH,25,8,'<>      B = (3)      E = (70) <>'
1457:      db      (-1) AND 0FFH,25,9,'<>      <>'
1458:      db      (-1) AND 0FFH,25,10,'<>      C = (38)      F = (73) <>'
1459:      db      (-1) AND 0FFH,25,11,'<>      <>'
1460:      db      (-1) AND 0FFH,25,12,'<>=====<>'
1461:      db      (-1) AND 0FFH,25,13,'<>      <ESC> Cancels Test <>'
1462:      db      (-1) AND 0FFH,25,14,'<>=====<>'
1463:      db      (-1) AND 0FFH,25,16,'TRACK? ...',bs,0
1464:
1465: me8:      db      (-1) AND 0FFH,24,5,'--- RADIAL ALIGNMENT CHECK ---'
1466:      db      (-1) AND 0FFH,10,9,'Away',(-1) AND 0FFH,36,9,'Spindle',(<
1467:      db      (-1) AND 0FFH,10,10,'13 12 11 10 9 8 7 6 5 4 3 2 1'
1468:      db      '-1 2 3 4 5 6 7 8 9 10 11 12 13',0
1469:
1470: me9:      db      (-1) AND 0FFH,25,2,'<>=====<>'
1471:      db      (-1) AND 0FFH,25,3,'<>      TRACK SELECTION <>'
1472:      db      (-1) AND 0FFH,25,4,'<>=====<>'
1473:      db      (-1) AND 0FFH,25,5,'<>      <>'
1474:      db      (-1) AND 0FFH,25,6,'<>      A = (35)      B = (44) <>'
1475:      db      (-1) AND 0FFH,25,7,'<>      <>'
1476:      db      (-1) AND 0FFH,25,8,'<>      C = (47)      <>'
1477:      db      (-1) AND 0FFH,25,9,'<>      <>'
1478:      db      (-1) AND 0FFH,25,10,'<>=====<>'
1479:      db      (-1) AND 0FFH,25,11,'<>      <ESC> Cancels Test <>'
1480:      db      (-1) AND 0FFH,25,12,'<>=====<>'
1481:      db      (-1) AND 0FFH,25,14,'TRACK? ...',bs,0
1482:
1483: me10:     db      (-1) AND 0FFH,23,5,'--- DISKETTE CENTERING CHECK ---',0
1484:
1485: me11:     db      (-1) AND 0FFH,23,14,'Press <SPACE BAR> For New Track',0
1486:
1487: me12:     db      (-1) AND 0FFH,24,5,'--- AZIMUTH ALIGNMENT CHECK ---'
1488:      db      (-1) AND 0FFH,25,14,'<ESC> Will Cancel AZIMUTH Check',0
1489:
1490: me13:     db      (-1) AND 0FFH,25,2,'<>=====<>'
1491:      db      (-1) AND 0FFH,25,3,'<>      TRACK SELECTION <>'
1492:      db      (-1) AND 0FFH,25,4,'<>=====<>'
1493:      db      (-1) AND 0FFH,25,5,'<>      <>'
1494:      db      (-1) AND 0FFH,25,6,'<>      A = (0)      B = (76) <>'
1495:      db      (-1) AND 0FFH,25,7,'<>      <>'
1496:      db      (-1) AND 0FFH,25,8,'<>=====<>'
1497:      db      (-1) AND 0FFH,25,9,'<>      <ESC> Cancels Test <>'
1498:      db      (-1) AND 0FFH,25,10,'<>=====<>'
1499:      db      (-1) AND 0FFH,25,12,'TRACK? ...',bs,0
1500:
1501: me15:     db      (-1) AND 0FFH,15,8,'#####'
1502:      db      (-1) AND 0FFH,15,12,'#####'
1503:
1504: me19:     db      (-1) AND 0FFH,27,5,'--- SPINDLE SPEED CHECK ---'
1505:      db      (-1) AND 0FFH,25,14,'<ESC> Will Cancel RPM Check',0
1506:
1507: me20:     db      ' Milliseconds',0
1508: me21:     db      ' RPM',0
1509:
1510: BLANKS:   db      '
1511:      db      ',0
1512:
1513: SELMSG:   db      (-1) AND 0FFH,15,12,'WHICH DRIVE HAS THE DDD - A,B,C,D ?
1514:      db      (-1) AND 0FFH,25,14,'SELECTION ? ....',0
1515:
1516:
1517: *****
1518: *** "POSITION CURSOR" ***
1519: *****
1520:
1521: set$cur db      0ffh

```



```

1522: column ds      1      ; set column
1523: row      ds      1      ; set row
1524:          db      0      ; terminator
1525:
1526: *****
1527: *** "PROGRAM VARIABLE STORAGE" ***
1528: *****
1529:
1530: temp ds      2      ; temp storage
1531: diff ds      1      ; 1st - 2nd read
1532: ctrk ds      1      ; current track
1533: savtrk ds     1      ; temp storage Hysteresis test
1534: dflag ds      1      ; seek delay flag
1535: point ds      2      ; table pointer
1536: xpoint ds     2      ; table pointer positive
1537: ypoint ds     2      ; table pointer negative
1538: xoff ds      1      ; positive offset
1539: yoff ds      1      ; negative offset
1540: bais ds      1      ; bais from test track (Hysteresis)
1541: hyerr ds      1      ; hysteresis error on first reading
1542: SELPT DS      1      ; STORES BYTE FOR DISC CONTROL PORT
1543:
1544: ;
1545: ;      Storage area for 16/24 bit divide routine
1546: ;
1547: msb ds      1
1548: msb1 ds     2
1549: msb2 ds     2
1550:
1551: inbuf: db      3      ; "max" byte for BDOS 10
1552: incnt ds      1      ; "cnt" buffer for BDOS 10
1553: buffer ds      4      ; console input buffer
1554: secbuf ds    256      ; sector buffer
1555:
1556:          ds    100      ; stack space
1557: stack equ     $      ; top down...
1558:
1559:
1560:

```

End Listing

Elegance
Power
Speed



C Users' Group
Supporting All C Users
Box 287
Yates Center, KS 66783

Circle no. 27 on reader service card.

Eco-C TM Compiler

Since November of last year, we've been testing our new Eco-C Compiler and now it's ready for your Z80™ CP/M™ system. Some of the features include:

- All data types, including float, double and long.
- Fast, efficient code. For example: Our versions of "seive" in January, 1983, BYTE; 15.8 seconds (standard) and 11.7 seconds (optimized).
- Uses Microsoft's MACRO 80™ for REL file output.
- Standard (K&R) file I/O and library (most in source) included.
- Easy assembly language interface.
- No royalty fees on generated code.

The price for Eco-C is \$350.00 and includes MACRO 80 (a \$200.00 value by itself). We'll also include a free copy of *C Programming Guide* while supplies last.

For further information, call or write:



P.O. Box 68602
Indianapolis, IN 46268
(317) 255-6476



Registered trademarks are: Zilog (Z80), Digital Research (CP/M), Microsoft (MACRO 80) and Eco-Soft (Eco-C).

Circle no. 37 on reader service card.

Building a Programmable Frequency Synthesizer

The Master Controller Board (MCB), a single-board computer available from Space-Time Productions, Chicago, Illinois, has an 8253 counter/timer on it that is used for a baud rate generator. Since it has two spare counter time channels, I decided to use them to construct a crystal-controlled programmable frequency synthesizer.

Phase-Locked Loop (PLL) Theory

A phase-locked loop is a very useful combination analog and digital device. It enables a digital frequency divider to multiply frequencies exactly. It performs this trick by controlling the frequency and phase of a voltage-controlled oscillator (VCO) in such a way that the VCO is locked into the frequency and phase of a reference oscillator. (See Figure 1, below.)

by Michael L. Simon

Michael L. Simon, Space-Time Productions, 2053 N. Sheffield, Chicago, Illinois 60614.

Copyright © 1983 Michael L. Simon. All rights reserved.

Permission is granted for non-commercial use only. Any commercial use without permission from the author is prohibited.

An example will help to explain the operation. A reference frequency of 1000 Hz is available; it is used as one input to a phase comparator. The other input comes from the output of the VCO through a programmable divider, which is set to divide by four. For the loop to be locked, the output of the programmable divider must be 1000 Hz. If the frequency is low, the phase comparator will generate longer pulses which, when averaged by the low pass filter, will create a higher voltage at the input of the VCO. This will cause the output of the VCO to increase in frequency, maintaining an output of 4000 Hz.

Similarly, if the output of the VCO is above 4000 Hz, the phase comparator will generate shorter pulses. This will reduce the frequency, again restoring the original 4000 Hz. Thus a phase-locked loop and a divide-by-four circuit are able to multiply an input frequency by four. If the input to the reference of the loop has high stability, the output of the loop will also tend to have high stability.

Figure 2 (page 81) shows the circuit diagram of a phase-locked loop.

The 4046 CMOS Phase-Locked Loop Chip

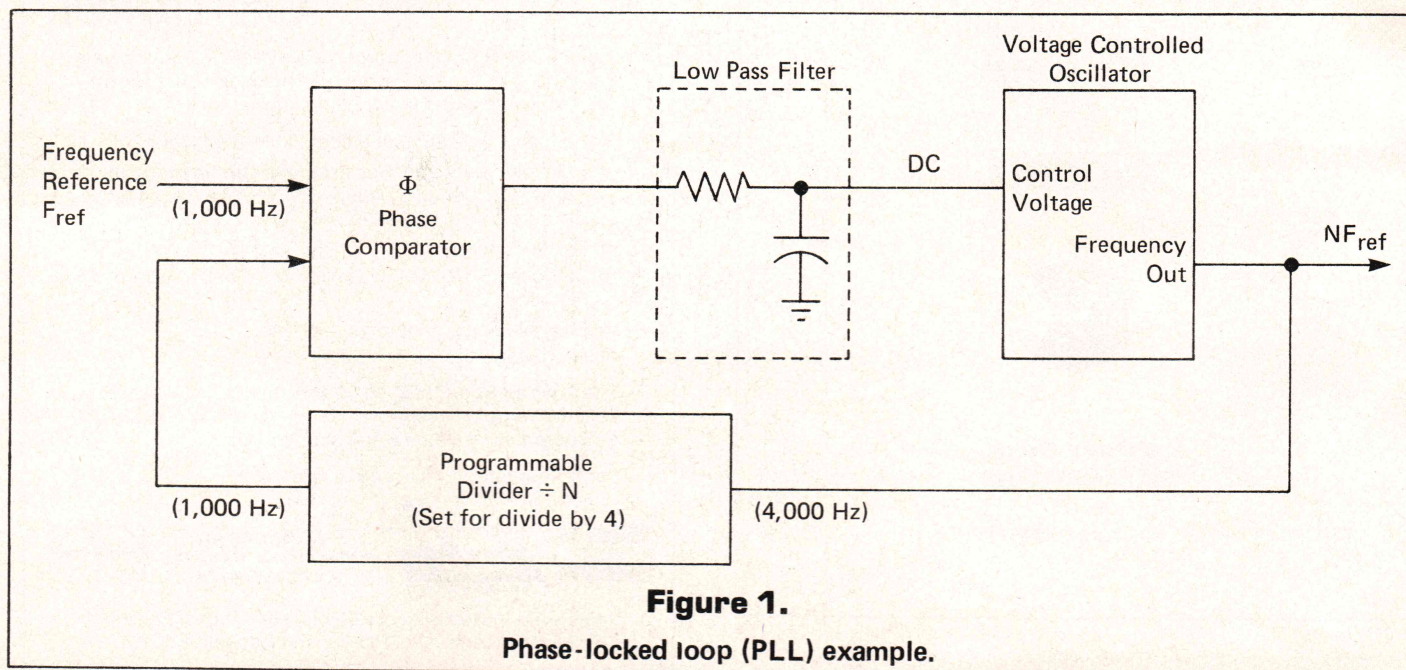
The 4046 is a CMOS phase-locked loop which is fairly inexpensive and oper-

ates over a wide frequency range (from less than 1 Hz to over 1 million Hz). The 4046 has a phase detector that is edge sensitive. This is useful when the signals to be synchronized are not perfect square waves. The chip has a second phase detector that can be used to tell if the loop is out of lock. Because the 4046 is a CMOS device with high input impedance, loop filters with high value resistors and low value capacitors are possible. This makes possible low-frequency loop filters built from small, inexpensive capacitors.

The Single-Board Computer

The MCB is a Z80-based, single-board computer with many useful features. Among these features are 72 parallel I/O lines, two serial I/O ports, and two counter/timer chips with a total of eight counter/timers. Also available for the board is a version of the TDL monitor, modified to allow control of the programmable dividers of the 8253. This program makes frequency synthesizer experiments a snap since it has commands that directly control the dividers in the 8253 counter/timer chip.

One channel of the 8253 is used for the on-board baud rate generator. Each channel of the baud rate generator is set up for a different baud rate; three rates



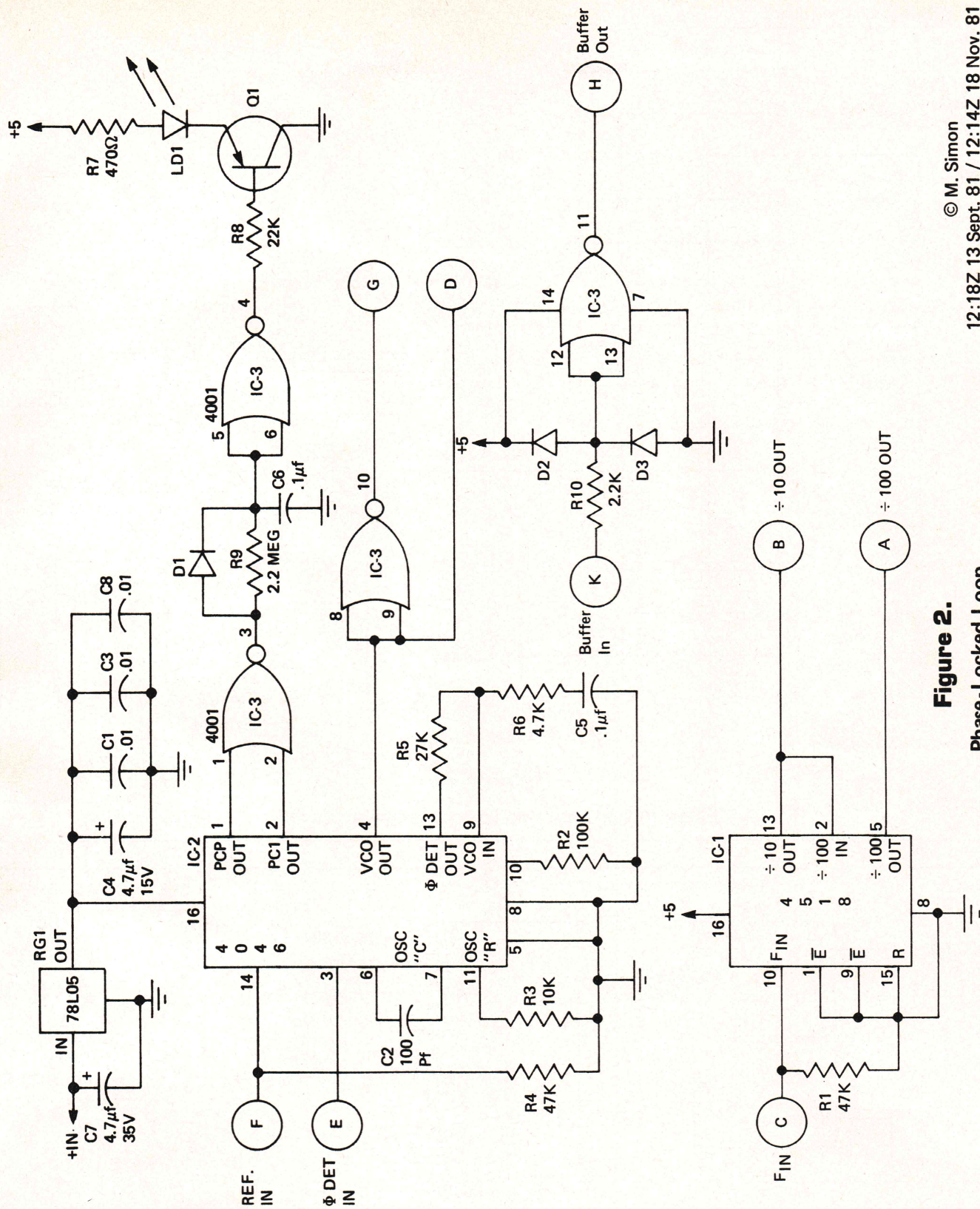
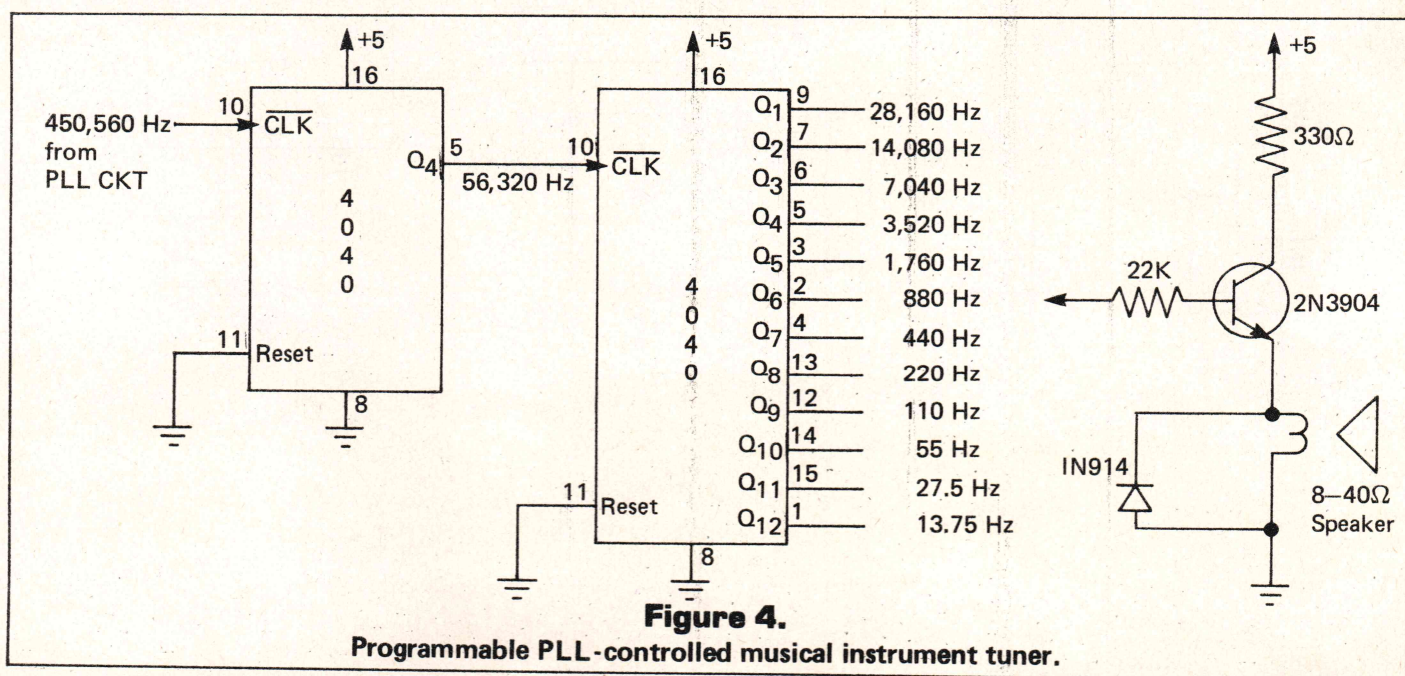
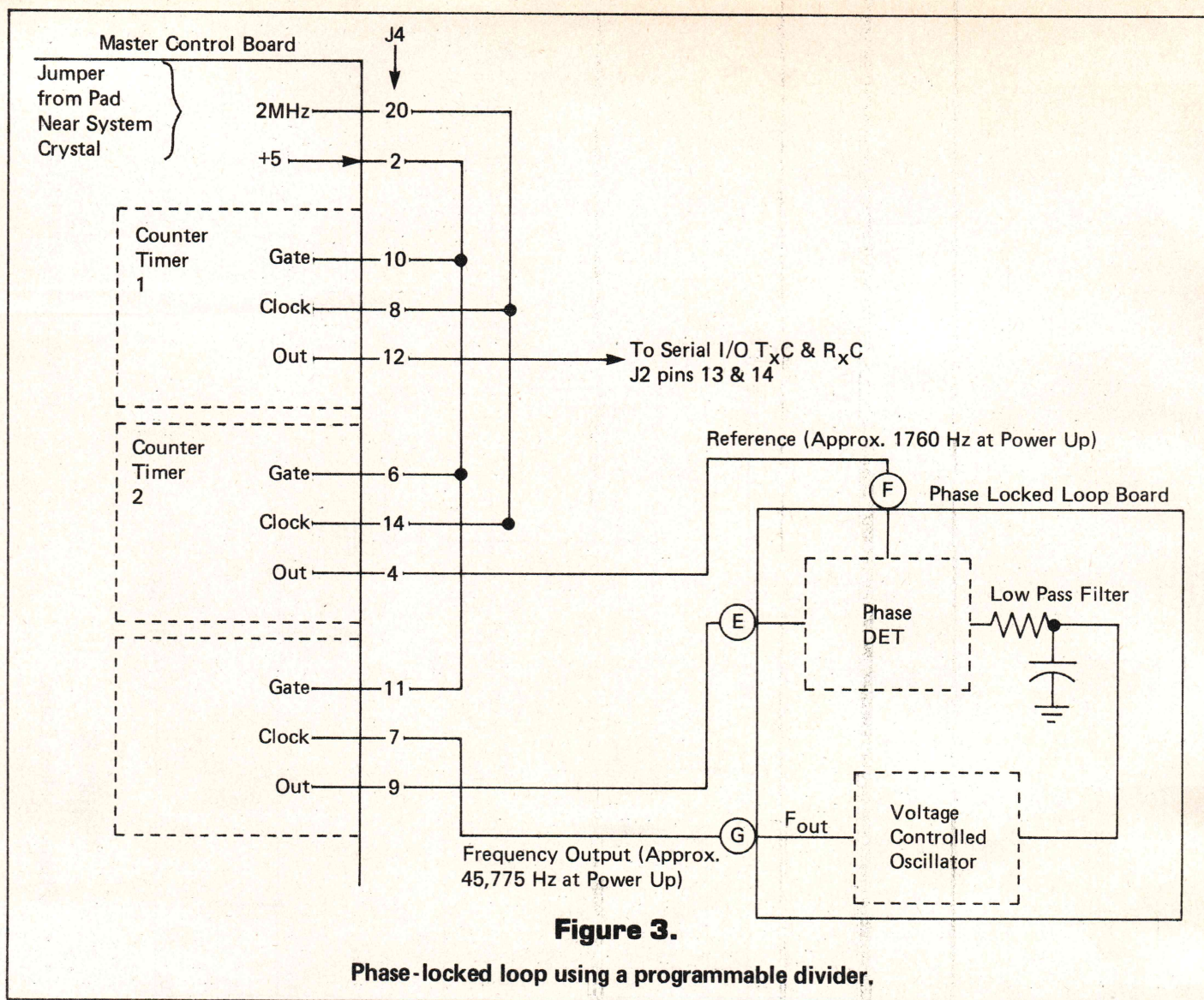


Figure 2.
Phase-Locked Loop.

© M. Simon

12:18Z 13 Sept. 81 / 12:14Z 18 Nov. 81



(110, 300, and 4800) are available. Since I use 300 baud to talk to the MCB, which uses channel 1 of the 8253, channels 0 and 2 are available for experiments. The 8253 is set up by the monitor program to act as a programmable divide-by-N circuit and to produce a square wave. If N is an even number, the output will be a square wave. If N is an odd number, the output will be approximately a square wave with one half of the cycle longer than the other half; refer to Figure 3 (page 82) and Listing One (page 88).

The gate of each channel used must be pulled high for the 8253 to perform the divide function. A 2 MHz buffered, crystal-controlled reference signal is available from the CPU's clock generator. This signal is required by the baud rate generator and is brought to pin 20 of header J-4 from the pads near the crystal by a jumper wire soldered to the underside of the board. From this pin I used reusable jumper wires on the header pins to bring the 2 MHz signal to the inputs of counters 0, 1, and 2. These jumper wires are about the handiest I have ever used on .025 square pins. They make wrapped wire methods seem very clumsy, especially when doing experiments, since the socketed jumpers are easy to move and reuse.

After hooking up the PLL according to diagram 2, I expected an output frequency of about 45,775 Hz—I say “about” because the output frequency depends on the exact frequency of the CPU time base generator and the divide ratios chosen for the baud rate generators. My oscillator was slightly low, so I got a different value.

I prepared a chart of divide ratios versus hexadecimal numbers. The baud rate generators could be programmed to recognize decimal values. However, by using hexadecimal numbers, over six times the range of divisors is possible.

My first experiment was to get the system to go out of lock on the low end. The frequency at which this happens depends on the components used in the loop filter. With the components chosen, the loop goes out of lock at a reference frequency of about 200 Hz into the phase comparator. The exact frequency will vary with the rate at which the PLL has to change between new inputs. The loop goes out of lock on a jump from divide-by-1000 hex to divide-by-2000 hex on the channel 2 counter. However, if you can inch up to 1F00 and keep the loop in lock, it is possible to jump from there to 2000 hex and still keep the loop in lock.

The same method is used for getting the loop back in lock from an out-of-lock condition. To get the loop to respond to lower frequencies and still stay in lock, try increasing R5 and C5. (Check out *Design of Phase Locked Loop Circuits*

With Experiments by Howard M. Berlin, Howard W. Sams Inc., for methods of determining the best values for the loop filter.)

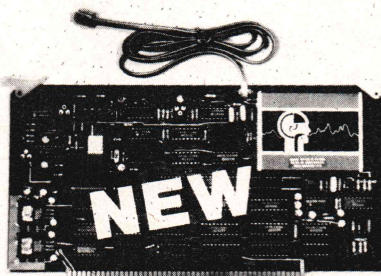
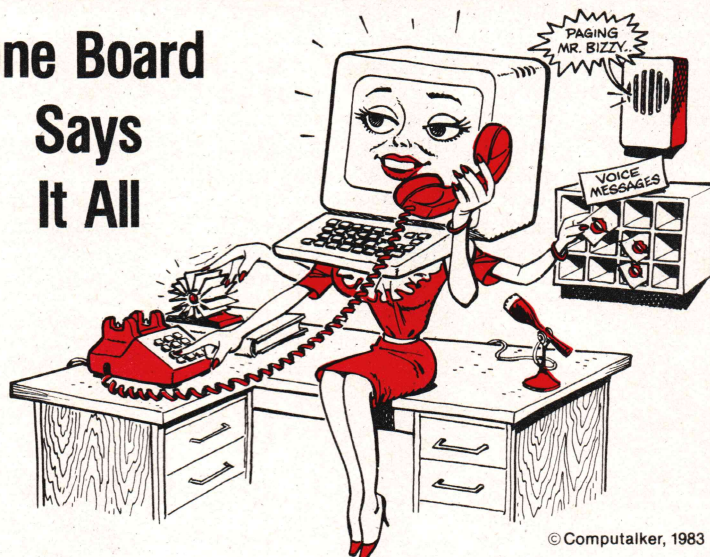
On the high end it is possible to keep the loop in lock until well past 1 MHz. However, I have had some PLL's lock up at the high frequencies. This is probably due to the speed limitation of the chips I used. By removing power and beginning again, everything was returned to normal.

Solid State Scientific of Montgomery, Alabama, makes a version of the 4046 with a VCO guaranteed to oscillate up to

5 MHz. The part is called a 4446. By reducing capacitor C2 to 50 pf, I got mine to stay in lock past 2.5 MHz. For those who want to go past 2.5 MHz, I recommend using about 20 pf at C2. You must also use a divider capable of higher frequency input than the 8253. Fortunately, the Master Controller Board has a socket for the AMD 9513 counter/timer, which is guaranteed up to 7 MHz.

It is possible to generate almost any frequency with the right choice of dividers. For example, I wanted to tune up my AM radio. To do this I needed to tune up the

One Board Says It All



IEEE-696 / S-100 BUS

TELEPHONE INTERFACE

- FCC Approved
- Initiate & Answer Phone Calls
- Touch-Tone® Generator & Decoder
- Detect Phone Line Status (busy, ringing, answered, etc.)

VOICE DIGITIZER

- Record Speech
- Speech Storage in RAM or Disk
- Speech Output
- Choice of 5 Bit Rates

CompuFone: COMPUTER INTERACTION BY TELEPHONE

Access and enter data to your computer from any telephone via the Touch-Tone® keys. Verify your transactions by recorded speech. So many exciting applications for the CompuFone are possible: voice mail, telephone answering machine, electronic funds transfer, telephone soliciting and data gathering, electronic Rolodex®, data base query with voice output, order entry and inventory control, telephone banking—and that's just the beginning.



COMPUTALKER, 1730 21st St., Santa Monica, CA 90404
Call us at (213) 828-6546

Circle no. 20 on reader service card.

intermediate stages first. This requires a frequency of 455,000 Hz. With the crystal in my board, loading counter 0 with 3A4 hex and counter 2 with 1000 hex gave me a readout on my frequency counter of 454,995.0 Hz. Not bad, but I wanted to get closer.

The first step was to determine the frequency of the clock on the MCB. Checked with my trusty calculator, 3A4 converted to decimal is equal to 932; 1000H is 4096. Dividing 454,995 Hz by 932 found the reference frequency. Multiplied by 4096, the result was the crystal-controlled input clock frequency. My result was 1,999,634.6 Hz.

The dilemma now was to determine which divider ratios would yield the correct output frequency. There is a formula for doing this. Unfortunately, I had long since forgotten it, so I invented a trial-and-error method to determine the correct divisors. Since computers are much faster at trial and error than humans, I wrote a BASIC program (see Listing Two, page 91) to handle it. If I had known the right algorithm, there would have been quicker results, but the program does the job.

The main idea is that the phase-locked loop may be used to generate any desired frequency to any degree of accuracy by choosing the right values for the controlling dividers. By using this program, I came up with values of 808 for divider #0 and 3551 for divider #2. This gives an accuracy of 0.13 ppm. I converted 808 and 3551 to hexadecimal (328h and 0DDFh) and entered the values in the 8253. 455,000.0 Hz was the result!

With suitable dividers following the loop output, audio tones can be generated. These could be used to fine tune your favorite string instrument (see Figure 4, page 82). The 4040 circuit gives output ranging from the lowest to the highest octave, evenly spaced at octave intervals. The even-tempered scale used in most musical instruments uses twelve notes per octave interval. Each note is higher in frequency than the preceding note by a factor of 1.0594631. Table I (below) gives a series of frequencies that facilitates tuning all the notes in any octave. The beauty of this set-up is that with the proper choice of dividers, any scale may be reproduced, regardless of the number of notes or the size of the interval.

The Phase-Locked Loop board is available as a kit for \$28.75 postpaid from Space-Time Productions, 2053 N. Sheffield, Chicago, Illinois 60614; (312) 327-0391. The Master Controller Board is available bare for \$54.90 with postage. The customized TDL Monitor is available for \$69.90. A partially populated MCB, assembled and tested, running Tiny Controller Basic, is \$304.90 with postage.

DDJ

(Listing begins on page 88)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 201.

Note	Ratio	Input to 4040 Divider	A=440
A	1.0000000	450560.00 Hz	440.00000 Hz
A#	1.0594631	477351.69 Hz	466.16375 Hz
B	1.1224620	505736.47 Hz	493.88327 Hz
C	1.1892071	535809.15 Hz	523.25112 Hz
C#	1.2599210	567670.00 Hz	554.36523 Hz
D	1.3348398	601425.42 Hz	587.32951 Hz
D#	1.4142135	637188.03 Hz	622.25393 Hz
E	1.4983071	675077.20 Hz	659.25507 Hz
F	1.5874010	715219.44 Hz	698.45648 Hz
F#	1.6817928	757748.56 Hz	739.98882 Hz
G	1.7817974	802806.63 Hz	783.99084 Hz
G#	1.8877486	850544.00 Hz	830.60937 Hz
A	2.0000000	901120.00 Hz	880.00000 Hz

Table 1
Twelve Tone Scale Frequencies

EP-PLOT[®]

EP-PLOT: The Plot Program that is also a Plotter

EP-PLOT converts your MX-80 or FX-80 with GRAPTRAX into a high resolution digital plotter with the same command language as a Houston Inst. DM/PL+ plotter plus much more. Produce high resolution plots without an expensive plotter. If you have CP/M-80, an MX-80 or FX-80, with EP-PLOT, you can now have full plotting capability for \$180.00.

EP-PLOT \$180.00

Demonstration Disk \$15.00

(refundable with system purchase)

Available for CP/M in 8" SSD and 5" NorthStar SSD

Order Now — Send Check or Money Order to:



3138 Via Loma Vista, Escondido, CA 92025
Mail to: P.O. Box 1825, Escondido, CA 92025
(Calif. residents add 6% sales tax)

GRAPTRAX is a reg. trademark of EPSON-AMERICA Inc.
DM/PL+ is a reg. trademark of Houston Instrument Inc.
CP/M is a reg. trademark of Digital Research Inc.

Circle no. 100 on reader service card.

TALISMAN^{T.M.}

It's almost magic!

CP/M[®] Terminal Translation & Multiple Keyboard Redefinition Program

- ★ Emulates ANY interactive terminal — runs ANY CP/M 2.2 software on your microcomputer.
 - ★ Bridges between your micro and mini or mainframe terminals.
 - ★ Redefines any key(s) to produce up to 249 characters.
 - ★ Creates, saves, edits, and retrieves up to 255 keyboard overlays.
 - ★ Reprograms *on-the-fly*, while you're running another program.
 - ★ And so much more!
 - ★ A godsend for programming, data entry, word-processing.
 - ★ Transparent to the user. No wiring or soldering.
- User-friendly. Best documentation in the industry.

TALISMAN[™] is only \$125, manual included.

California residents add 6% sales tax.

Call or write for free details.



DISCO-TECH[®]
a division of Morton Technologies, Inc.
600 B St. / P.O. Box 1659
Santa Rosa, CA 95402
Tel. 707 / 523-1600
Dealer inquiries invited.

Circle no. 34 on reader service card.

SOURCE CODE

I N C L U D E D

B + TREE^{*}

The **SOFTWARE 99[™]** B + TREE indexing system is a BASICA Source Code implementation of file maintenance procedures for record modification, insertion, deletion and query. Featured is the description of integration procedures necessary to couple the B + TREE with application programs.

CROSS REFERENCE/ DOCUMENTATION UTILITY^{*}

The need for a structured program listing & cross reference is satisfied using the **SOFTWARE 99[™]** CROSS REFERENCE/ DOCUMENTATION UTILITY. Line numbers and variable names are cross referenced against their usage within a BASICA program. The listing is in an easily read form with one statement per line and structuring of IF statements and LOOPS. Output can be selected to the screen, printer or ASCII file.

GENERAL LEDGER^{*}

The **SOFTWARE 99[™]** GENERAL LEDGER is a remarkably easy to use, but fully integrated system of 9 BASICA programs run from a main menu. Some of these programs include Account Maintenance, Transaction Entry, Trial Balance, Balance Sheet and Income Statement. For those interested in Source Code, the 9 programs are all written in the same style, so that a study and understanding of one program gives you an understanding of each program.

^{*}MINIMUM SYSTEM REQUIREMENTS:
64K, PC-DOS or MS-DOS
(any version), 1 disk drive

ONLY **\$99** EACH

Please add \$6 for postage and handling.



Calif. residents add 6% tax



SOFTWARE 99[™]
THE SOURCE FOR SOURCE CODE

702 Washington Street #174
Marina del Rey, CA 90291

PHONE ORDERS (213) 399-7463 24 HOURS

PC-DOS a trademark of IBM Corporation
MS-DOS a trademark of Microsoft Corporation

Circle no. 95 on reader service card.



COMPUTER RESOURCES

OF WAIMEA

Menu Driven Programs running under SPELLBINDER

DearJohn



will free your operators from entry errors. Completely menu driven Mailing List Management Programs with thoughtfully formatted screens to ease the entry of addresses and names. Integrity of the data file created is automatically verified. No more printing 'bad' mailing labels with misplaced data; AND each address record can be individually classified to allow inclusion (or exclusion) when printing labels or form letters.

(10 Programs) **\$89.50**

SBMenu SBHelp

make your version of Spellbinder menu driven. No more fumbling through manuals for obscure or little-used commands. The complete instruction set can be displayed at any time without disturbing the text on the screen. These two programs put Spellbinder into the world class of word processors, allowing the user to easily use its truly outstanding features.

(With two help files) **\$55.00**

AutoSave

automatically saves the text on the screen to a previously named file. Prevents the inadvertent but all so common save to a wrong name. Works with source code and program code as well as with text files. All instructions on disk.

(3 Programs) **\$97.50**

Send Check, Money Order, Visa, Master Charge:

COMPUTER RESOURCES OF WAIMEA
P.O. Box 1206
Kamuela, Hawaii 96743
Phone: (808) 885-7905

Circle no. 24 on reader service card.

ADVERTISE IN THE
Dr. Dobb's Journal
SPECIAL
TELECOMMUNICATIONS
ISSUE

Coming this February

Space Reservations Deadline:

December 1st, '83

Material Deadline:

December 8th, '83

Contact:

Walter Andrzejewski, Beatrice Blatteis
or Alice Hinton
for assistance today!
(415) 323-3111

Dr. Dobb's Journal
P.O. Box E
Menlo Park, CA 94026



- ★ Press a key and **MagiKey™** will transmit a character string to a program, execute a sequence of programs automatically (no need for CP/M SUBMIT or XSUB), or do just about anything else you wish within a program or at the CP/M level. Gives new meaning to one key, several keys, or your whole keyboard.
- ★ Exclusive **MagiKey™** features include strings with "fill-the-blanks" input prompts, nested key definitions, string and keyboard output redirection, arbitrary string size, advanced string editing features, and more . . .
- ★ **MagiKey™** is completely transparent, installs in seconds without hardware or software modifications, is compatible with all popular application programs and languages, and is available for all 8080-8085-Z80 microcomputers using CP/M 2.2.
- ★ **MagiKey™** has true macro processing capabilities and much more powerful features than any ordinary keyboard redefinition program . . . there is no comparison.

\$10000 **postpaid**

8" SSDD Standard, inquire about 5 1/4"
check, VISA, M/C
add 6% tax in CA
CP/M (TM) Digital Research, Inc.



microSystems
16609 Sagewood Lane
Poway, California 92064
(619) 693-1022

Circle no. 82 on reader service card.

MEGA-BYTE

SPECIALIZING IN HARD DISKS AND HIGH PERFORMANCE STREAMING TAPE UNITS

MD-10 11 MEGABYTE (FORMATTED) 5 1/4" HARD DISK Including **POWER!** \$1995

POWER!

Check out InfoWorld's Rave review of POWER Nov. 8/82)

55 CP/M UTILITIES IN ONE PACKAGE

...The super program that puts you in control of CP/M (or MS-DOS) and your winchester.

POWER automatically numbers disk files. Just pick file number to Copy, Erase, Reclaim, Type, etc.
...Your computer feeds the file names automatically. You do NO typing! NO typing error ever!
YOU DON'T NEED SYSTEM DISK IN ANY DRIVE.
No more BDOS ERROR!

YOU Test and Fix bad disks! Reclaim accidentally erased files or programs! Single step thru memory up or down! Search, view, change memory or disk in a snap! See Status and File Size instantly! Verify check-sums for programs! Load or Save programs at any address.

55 prompted user-friendly functions for housekeeping and a 120 page easy-read user's guide make POWER your most often used software. You'll use it every day!

Now, POWER! also includes a special program that lets you lock sensitive files, so that only you can access them. Without the secret PASSWORD which you can create and change at will, no prying eyes will ever know your secret file even exists.

SOME MAJOR POWER COMMANDS:

REN	CM	WRITEGR	DISK	STAT
TYPEA	USR 2	DUMPA	SPEED	SETDIR
SIZE	ERA	SEARCH	WRITE	RECLAIM
XUSER	TYPEX	?	DUMPX	DS
SETRO	CHECK	COPY	FILL	READ
GROUP	TEST	TYPEH	EX	DUMP
SAVE	SETWR	EXIT	DIR	MOVE
READGR	LOG	USER	TYPE	JP
DUMPH	LOAD	SETSYS	RUN	

For this low price your winchester will be delivered completely assembled and tested, with drive controller, case, power supply, cabling, Z-80 interface and the best disk controller software on the market.

- The unique and simple interfacing system does not tie up existing ports.
- Significantly faster than other winchester subsystems which interface through the IEEE-488 port.
- CP/M drivers require minimum memory overhead (about 2K-other systems require as much as 6k)
- The MD-10 can read or write a 64K file in less than four (4) seconds.
- A network system will be available shortly which can support up to sixteen mixed types of computers from one MD-10 or larger disk subsystem.
- With POWER! Software files can be code word protected.
- Other systems available: 22 and 44 megabytes

MD-20 with 22MB formatted: \$2895

MD-44 with 44MB formatted: \$3995

- Up to eight (8) winchester subsystems can be interfaced to one computer.
- Software supports 32 different user areas per MD-10.
- Backing up hard disk files is simple with the special software which is provided with all subsystems.
- With a single hard disk installation, the MD-10 subsystems becomes units A and B with the standard drives being designed E and F if a second MD-10 is installed later it becomes units C and D.
- MD-10 or larger systems will interface with IBM PC or any Z-80 computer (CCS, APPLE (CP/M), ZENITH/HEATH, NORTHSTAR, GODBOUT, XEROX 820, Z-80/S100, ALSPA, or TRS-80 MOD II) using CP/M, OASIS, PCDOS 2.0, and Godbout Software supports both CP/M and MP/M816.
- Double density modifications are available allowing you to later increase the capability of an MD-10 to about 20 megabytes, MD-20 to 40, etc.
- Full One-Year Warranty

HIGH PERFORMANCE STREAMING TAPE

Unique! Simple! Compatible!

- Field-proven Archive Sidewinder tape unit.
- Interfaces with ANY SASI/SCSI compatible hard disk peripheral.
- Full CRC error-checking
- Superb reliability.
- Full One-Year Warranty.
- Fast (up to 20MB in less than 5 minutes, up to 40MB in less than 10 minutes).

\$1795.00 20MB

\$1995.00 45/60MB

\$29.95 3M DC300XL

Streaming Tape Cartridges

MEGA-BYTE

ORDER
Name

Qty Price Total

VISA ☐ M/C ☐ COD ☐

Total Order

Card # _____ Exp. Date _____ If Utah Tax

Signature _____ Total Enclosed



TO ORDER OR FOR MORE INFORMATION:
(801) 257-7033 or mail to MEGA-BYTE INC. RT. 3 Box 35, Tremonton, Utah 84337
TERMS: Cashiers Check, VISA, M/C or COD.
Shipping charges added to all orders.



NAME _____
COMPANY _____ TITLE _____
ADDRESS _____ CITY _____
STATE _____ ZIP _____
TEL () _____

CP/M is a registered trademark of Digital Research. OASIS is a proprietary product of Phase One Systems Inc. Z-80 is a trademark of Zilog. MD-10 is a registered trademark of Media Distributing.

Frequency Synthesizer (Text begins on page 80)

Listing One

; 11:58z 22 MARCH 83 CT FOR PLL ART

CDL's Z80 Macro Assembler C12012-0312

Modified for MASTER CONTROLLER BOARD 21 MARCH 83 M.Simon

; EQUATES FOR THE 8253 TIMER CHANNELS

006C	CTR0	== 6Ch	;8253 CHAN 0
006D	CTR1	== CTR0+1	
006E	CTR2	== CTR0+2	
006F	CTR3	== CTR0+3	;8253 CONTROL ; CHANNEL

;THE "BAUD" PROGRAM GETS TWO PARAMETERS OFF THE
; STACK. THE FIRST OFF THE STACK IS A TWO BYTE
; HEX NUMBER THAT IS THE DIVIDER THAT IS TO BE
; USED BY THE COUNTER. THE SECOND NUMBER IS USED
; TO DETERMINE THE COUNTER TO BE USED.

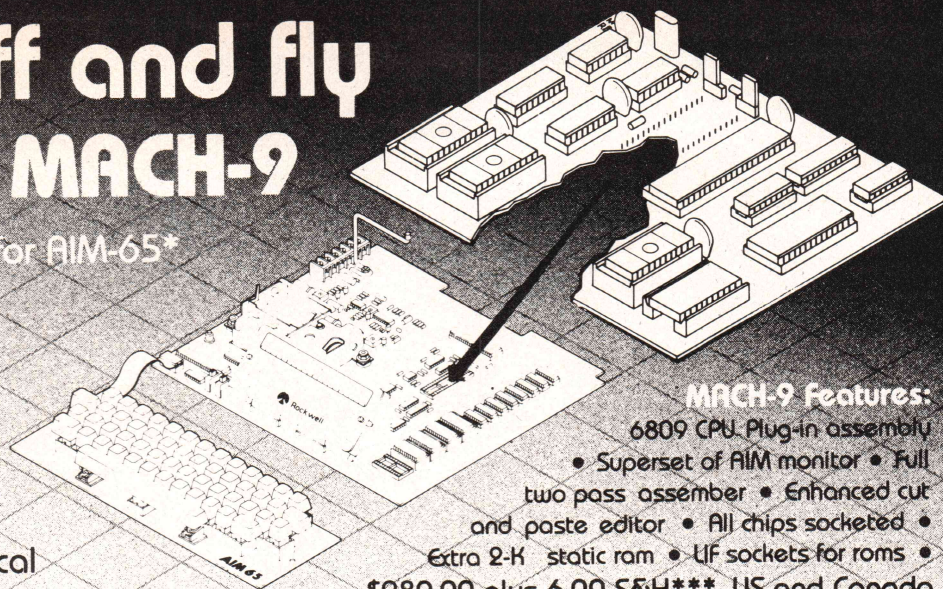
;DIVIDERS CAN BE FROM 2 TO 0FFFFh, COUNTERS
; ARE 0,1, & 2
;

017A	CD 065E	BAUD:	CALL	EXPR	;GET 2 PARAMS ; & PUT THEM ON ; THE STACK ;BAUD RATE ;COUNTER # ;TEST FOR CTR #
017D	D1		POP	D	
017E	E1		POP	H	
017F	7D	BRI:	MOV	A,L	
0180	B7		ORA	A	
0181	2809		JRZ	OUTC0	
0183	3D		DCR	A	
0184	2811		JRZ	OUTC1	
0186	3D		DCR	A	
0187	2819		JRZ	OUTC2	
0189	C2 057F		JNZ	ERROR	
018C	3E36	OUTC0:	MVI	A,00110110B	
018E	D36F		OUT	CTR3	
0190	7B		MOV	A,E	
0191	D36C		OUT	CTR0	
0193	7A		MOV	A,D	
0194	D36C		OUT	CTR0	
0196	C9		RET		
0197	3E76	OUTC1:	MVI	A,01110110B	

(Continued on page 90)

Take off and fly with the MACH-9

The 6809 adaptor for AIM-65*



"Just Released"

MACH-9 Control Pascal

A superset of standard Pascal

No rom expansion board necessary

Introductory Price \$69.00 plus \$5.00 S&H US and Canada

In Europe contact:

SKYLAB s.r.l. • Via M. Gioia 66
20125 Milano Italy • Tel. (02/688.38.06)
Telex 335535 SKYLAB I

MACH-9 Features:

6809 CPU Plug-in assembly
• Superset of AIM monitor • Full
two pass assembler • Enhanced cut
and paste editor • All chips socketed •
Extra 2-K static ram • LIF sockets for roms •
\$289.00 plus 6.00 S&H*** US and Canada

For more Information Contact:

Modular Mining Systems, Inc. • 1110 E. Pennsylvania St.
Tucson, Arizona • 85714 • (602) 746-0418

In the UK Contact:

RCS Microsystems Ltd. • Gresham House
Twickenham Rd. • Feltham Middlesex
TW13 6HA • 01-898-3775.



*AIM-65 is a trademark of Rockwell International

**Byte Magazine Sept. 1981 pg. 192

***\$20.00 S&H for overseas.

Circle no. 70 on reader service card.

Train Your Computer to DO MORE with 60 UNIX-Style Tools from Carousel.



YOU TRAIN A COMPUTER by writing scripts for it to follow at your command. These scripts tell the computer the things you want it to do to complete a task. Once trained, your computer can repeat the script whenever you want. Carousel's manual and tutorials teach you how to write scripts.

CAROUSEL's SOFTWARE TOOLS add the power of 60 UNIX-style commands to your CP/M and MSDOS operating systems. The Tools work with your word processor, data-base, spread-sheet and other programs to improve their utility. The Tools are simple to use, easy to remember and useful for training computers.

ORDER TODAY: Carousel's Start-a-ToolKit is \$99, add-ons are from \$35. Buy the full Use-a-ToolKit for \$249; only \$100 more gets the source for most of the Tools.



CALL OR WRITE:



CAROUSEL MICROTOOLS, INC.

609 Kearney Street, El Cerrito, CA 94530 (415) 528-1300

UNIX is a trademark of Bell Laboratories; CP/M is a trademark of Digital Research. MSDOS is a trademark of MicroSoft. Carousel Start-a-ToolKit and Carousel Use-a-ToolKit are trademarks of Carousel MicroTools, Inc.

Circle no. 15 on reader service card.

Frequency Synthesizer

(Listing continued, text begins on page 80)

Listing One

```

0199      D36F      OUT      CTRTC
019B      7B        MOV      A,E
019C      D36D      OUT      CTR1
019E      7A        MOV      A,D
019F      D36D      OUT      CTR1
01A1      C9        RET

01A2      3EB6      OUTC2:  MVI      A,10110110B
01A4      D36F      OUT      CTRTC
01A6      7B        MOV      A,E
01A7      D36E      OUT      CTR2
01A9      7A        MOV      A,D
01AA      D36E      OUT      CTR2
01AC      C9        RET
    
```

```

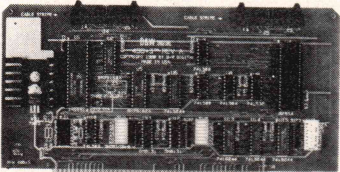
;THIS PROGRAM DOES THE POWER ON INITIALIZATION OF
; THE 8253 COUNTER TIMERS. THIS PROGRAM IS CALLED
; BY THE MAIN INITIALIZE PROGRAM. THE THREE
; COUNTER TIMERS ARE SET UP FOR BAUD RATE
; GENERATORS. CHAN. #0 = 4800 BAUD
; CHAN. #1 = 300 BAUD CHAN. #2 = 110 BAUD
    
```

```

01C8      CT53IN:  MVI      L,0          ;INIT CTR 0
01C8      2E00      LXI      D,26.        ; FOR 2Mhz CLOCK,
01CA      11 001A    CALL     BRI          ; 4800 BAUD

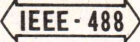
01CD      CD 017F    CALL     BRI          ; CTR 1
01D0      2E01      LXI      D,417.       ; 300 BAUD
01D2      11 01A1    CALL     BRI
01D5      CD 017F    CALL     BRI
01D8      2E02      MVI      L,2          ; CTR 2
01DA      11 0470    LXI      D,1136.     ; 110 BAUD
01DD      CD 017F    CALL     BRI
01E0      C9        RET
    
```

End Listing One



THE 488+3

IEEE 488 TO S-100 INTERFACE



- Handles all IEEE-488 1975/78 functions
- IEEE 696 (S-100) compatible
- MBASIC subroutines supplied; no BIOS mods required
- 3 parallel ports (8255A-5)
- Industrial quality; burned in and tested
- \$375

S-100

D&W DIGITAL
 20655 Hathaway Ave.
 Hayward, CA 94541
 415/ 887-5711

(Dealer inquiries invited)

Circle no. 33 on reader service card.

Listing Two

PROGRAM #1

```

10 X=3                ;TIMER 2 DIVIDE NUMBER
20 Y=3                ;TIMER 0 DIVIDE NUMBER
30 PRINT "INPUT CLOCK FREQUENCY":INPUT C
40 PRINT "          OUTPUT FREQUENCY":INPUT D
50 I=D/C              ;RATIO OF OUTPUT TO CLOCK
60 PRINT "INPUT ACCURACY IN PPM":INPUT A
70 A=A/1E6
80 Z= (I-(Y/X))/I
90 IF ABS (Z)<A THEN GOTO 1000
100 IF Z>0 THEN 600
110 IF Z<0 THEN 500
120 GOTO 1000
500 X=X+1:GOTO 80
600 Y=Y+1:GOTO 80
1000 PRINT"TIMER #2 DIVIDER" X,"TIMER #0 DIVIDER"Y
1010 PRINT"ACTUAL ACCURACY" Z
1020 GOTO 10

```

End Listing Two

STATEMENT OF OWNERSHIP, MANAGEMENT, AND CIRCULATION

(Act of August 12, 1970, Section 3685, Title 39, United States Code)

- Title of Publication: Dr. Dobb's Journal, Publication No. 307690.
- Date of Filing: October 26, 1983.
- Frequency of Issue: Monthly (12 issues, \$25).
- Location of Known Office of Publication: P.O. Box E, 1263 El Camino Real, Menlo Park, CA 94026.
- Location of Headquarters or General Business Offices of the Publishers: P.O. Box E, 1263 El Camino Real, Menlo Park, CA 94026.
- Names and Addresses of Publisher, Editor, and Managing Editor: Publisher, Jane Laidley, P.O. Box E, 1263 El Camino Real, Menlo Park, CA 94026. Editor, Reynold Wiggins, P.O. Box E, 1263 El Camino Real, Menlo Park, CA 94026. Managing Editor, Craig LaGrow, P.O. Box E, 1263 El Camino Real, Menlo Park, CA 94026.
- Owner: People's Computer Company, P.O. Box E, 1263 El Camino Real, Menlo Park, CA 94026.
- Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgagees or Other Securities: None.
- For Completion of Nonprofit Organization Authorized to Mail at Special Rates: The purpose, function, and nonprofit status of the organization and the exempt status for Federal income tax purposes have not changed during the past 12 months.
- Extent and Nature of Circulation:
 - Total number copies printed: Average number copies each issue during the preceding 12 months: 25,106. Actual number copies of single issue published nearest to filing date: 29,580.
 - Paid Circulation. 1. Sales through dealers and carriers, street vendors, and counter sales. Average number copies each issue during preceding 12 months: 6,567. Actual number copies of single issue published

nearest to filing date: 11,000. 2. Mail subscriptions. Average number copies each issue during preceding 12 months: 15,842. Actual number copies of single issue published nearest to filing date: 17,280.

C. Total Paid Circulation. Average number copies each issue during preceding 12 months: 22,409. Actual number copies of single issue published nearest to filing date: 28,280.

D. Free distribution by mail, carrier, or other means, samples complimentary, and other free copies. Average number copies each issue during preceding 12 months: 581. Actual number copies of single issue published nearest to filing date: 800.

E. Total distribution. Average number copies each issue during preceding 12 months: 22,990. Actual number copies of single issue published nearest to filing date: 29,080.

F. Copies not distributed. 1. Office use, left over, unaccounted, spoiled after printer. Average number copies each issue during preceding 12 months: 568. Actual number copies of single issue published nearest to filing date: 500. Returns from news agents. Average number copies each issue during preceding 12 months: 664. Actual number copies of single issue published nearest to filing date: 0.

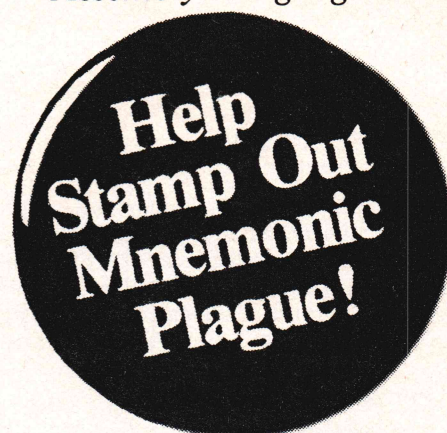
G. Total Average number copies each issue during preceding 12 months: 23,558. Actual number copies of single issue published nearest to filing date: 29,580.

I certify that the statements made by me above are correct and complete.

Jane Nissen Laidley

Jane Nissen Laidley, Publisher

End the Dark Ages of
Assembly Language....



with SMAL/80

SMAL/80	Assembler
HL=M (PTR) ;	LHLD PTR
DE=9;	LXI D,9
HL=HL+DE;	DAD D
IF A-L EQUAL	CMP L
THEN	JNZ L1
A=A-14	SUI 14
ELSE	JMP L2
A=L;	L1:MOV A,L
M(BC)=A;	L2:STAX B

SMAL/80 gives you the logical power, versatility and convenience of a compiled, structured high level language like Pascal, Ada or C, plus the efficiency of assembly language.

☐ intuitive, processor-independent symbolic notation system to make your programs easy to read, debug and maintain;

☐ programming constructs BEGIN... END, IF... THEN... ELSE, and LOOP... REPEAT, plus indentation, to graphically display the structure of your algorithms;

☐ extremely flexible macro and text pre-processor to create your own programming environment;

☐ compiler/linker to mix your input source code and relocatable object code, creating modular programs;

☐ translator program to automatically upgrade your assembly code to SMAL/80;

☐ available on CP/M disks with manual for \$150 plus \$4 shipping.

New! Z-80 version (runs on 8080's): \$175. 8080 version only: \$150. Macro-processor only: \$75. Available on CP/M disks. Add \$4 for shipping. Complete tutorial text: "Structured Microprocessor Programming" (Publ; Yourdon Press) \$20 plus \$2 shipping. Send for your free button and literature or try the Ultimate Demo: SMAL/80 is Guaranteed!

Chromod Associates,
1030 Park Ave., Hoboken, N. J. 07030
Telephone: (201) 653-7615

Also available from
WESTICO (203) 853-6880

Circle no. 17 on reader service card.

by Robert Blum

The column's a little short this month due to an accelerated deadline caused by CP/M '83. The interface series that I promised would start this month will be held off until next month. I apologize for the delay but I think you will enjoy it once it starts.

User Areas — Bah, Humbug

Several months ago, I cried on your shoulder about how old and decrepit my system had become. Well, since then, new disk drives have been installed, a new motherboard is now keeping a handle on the bus signals, and a 15MB hard disk happily gobbles up all the data I can feed it. Unfortunately, I should never have installed the hard disk because it pointed out to me one black hole that exists in CP/M: there is no easy way to partition logically related groups of data files.

Let me explain a little further. Right from the beginning I delegated my floppy disk drives to specific tasks. Drive A: is called the runtime drive. It contains only executable (.COM) files and any temporary work files that may be needed by the application programs. Drives B: and beyond are used for data files. I adopted this arrangement because it eliminates any confusion when switching from one application system to another. When I want to run word processing, the WP runtime disk goes into drive A:, and one of various WP data disks is mounted in a drive designated for data files. Very simple, very clean, and easy to manage.

Now comes the puzzle: how to allocate the hard disk so it is able to fit into my established scheme. My first attempt was to split the 15MB into two logical CP/M disks. The first one, drive C:, was 8MB in size with all remaining space given to drive D:. Merrily I copied disk after disk of data onto the Winchester. It's hard to describe my feeling of excitement as the data was gobbled up by an insatiable appetite. However, I became concerned after I ran my first directory listing. Displayed was such a hodge-podge of files that I knew something had to be changed.

I shuddered at my first thought — mapping a number of logical floppies onto the hard disk. Fortunately, it was only a fleeting thought. Even as I write this I can hear somebody scraping their fingernails across a blackboard.

User numbers would partition the files the way I wanted, but at the expense of duplicating a number of programs in each of the user areas. The overhead of multiple copies of the same program is

mandated by CP/M 2.2's inability to automatically switch user areas when searching for a file. For example, if you are currently in user area three, any file that is opened must also be in the same user area. This includes program overlays. One solution to the problem would be to use a CCP replacement such as ZCPR, but this would only provide a partial solution because an expanded search would only be available when ZCPR was resident. Once the executable program is loaded it has no way of knowing what user area it came from and hence no way of finding its overlays. Unfortunately, when Digital Research designed user areas, they failed to provide a means by which a file's FCB could communicate to the BDOS which user area it is in.

I don't mean to place all the blame on DR for this inadequacy. They provided a BDOS function that will switch the user area for any application program that is smart enough to use it. Unfortunately, none of the application developers decided to use it. Micro Pro took one step in the right direction by modifying WordStar to look at drive A: for its overlay files if they are not found on the currently logged-in floppy. This is better than nothing, but if the user area is not taken into account, it is of little use. Of course, at the time that CP/M 2.2 became available, disk capacities were relatively low and disks were typically swapped for each application. Today, however, hard disks are common and there is no excuse for having to allocate 15MB of space into logical floppies. As much as I hate to admit it, the standard version of CP/M 2.2 simply isn't suited to the hard disk environment.

It wouldn't seem too difficult to trap all the disk I/O BDOS calls and swap the user area for FCBs that contain a file extension of .COM or overlays that reside in the same user area. One problem with this occurs when an overlay file doesn't use .OVR for its extension. However, a little extra code could provide the desired results.

Converting to CP/M Plus would solve my problem. As a standard feature, it will search user area zero for any file that is not found in the current user area, although there are some restrictions. The target file must be marked with the \$SYS attribute, and it cannot be written to. This arrangement works well for executable programs, which is what I am complaining about, but for my money they don't get the prize. In designing CP/M Plus, DR decided

to add some very useful features. A file can be password-protected, and time and date stamping is available if your system supports a clock of some sort. To store this extra information, extended directory entries are recorded along with the regular directory entries.

Unfortunately, to cross-access user areas requires that the program manually switch the user area by calling the BDOS. I think a little extra work should have gone into allowing a program to pass an extended FCB address to the BDOS when a file is opened. This extended FCB would contain all the additional information necessary to interface to the new features that I have already mentioned, plus the user number. Better yet, instead of a user number, how about using an 8-character text name like the file name?

Making comparisons between different operating systems is usually like comparing apples to oranges, but I want to point out how files are handled on another system that I use daily. To protect the innocent I will call this new system Alpha. Alpha is a Z80-based computer with 64K of memory. The resident operating system requires approximately 10K, and the only transient part of the system is the operator interface, which is equivalent to CP/M's console command processor. The only place that my comparison begins to break down is that Alpha has a separate Z80-based processor that handles all disk I/O. I think my analysis still holds because CP/M Plus requires a minimum of 96K if you want to use the extended features. In Alpha's case, this would also be enough to hold all the disk logic.

A file specification on Alpha is contained in two control blocks. The first is called a "file control block" and is used for various pointers and general parameter passing. A two-byte area of the FCB points to another control block, the "file name block." Contained in the FNB are all the file's attributes in text form. Included are eight characters for the file set name (equivalent to user area), eight characters for the file name, and eight more characters for the file extension. Following this information is a complete description of the file. This includes record length, read and write passwords, and some other control information which isn't germane to this discussion. When the program wants to open a file, the FCB address is loaded into a register and a command code is placed into another. A call is made to the operating system through a restart

instruction, and away you go.

If it sounds simple, it is. Take, for example, the command line necessary to assemble a program:

ASM BLUM-PROGRAM.ASM

A few moments later the BLUM-fileset contains the assembler's output even though I was logged into the TEMP-fileset.

I have worked on a number of computers over the years and Alpha is my favorite. The real surprise comes when one considers Alpha's origin. Until Alpha's operating system was far enough along in the development cycle to stand alone, it was run under CP/M. Even after a few years of refinement, there are a number of CP/M's operational features still being used. Of course, I understand that a lot of things can be done when designing a new system that can't be done when modifying an existing one because of compatibility problems. Still, I think a lot can be accomplished if the motivation exists. I am sure that this discussion won't teach DR anything new; I have met a few of their people and they are sharp. But like so much else in this business, decisions are based on market potential. And as long as the semiconductor manufacturers continue to bring out new and improved processors every couple of years, there is little time left to do anything but convert the existing systems. Well, as I have said before, that's the state of the art.

A Point Well Taken

As I was putting the final touches on this month's column, I received a note from a reader complaining about the apparent lack of proofreading of the twittle subroutine that I ran a couple of months ago. He found the word "asterisk" misspelled in the comments describing the program. I must admit that my first impression was that he was being rather nit-picky, but after a moment's thought (it takes that long for my defense mechanism to readjust), I decided that he had a valid point. To the eyes of many, words are like notes to the ears of a musician. I am not an artist, but I am aggravated by a fuzzy radio station or a scratched record; my reader found a misspelled word to be offensive to his eyes. I apologize for the inexcusable error and will pay more attention to detail in the future.

DRI Patch 3 — SUBMIT.COM

Products Affected: CP/M 2.2

Copyright © Digital Research, Inc. 1982. All rights reserved. Used with permission of Digital Research, Inc. Text adapted from the original.

Error Description: If drive A is not the default drive when you run the SUBMIT program, the \$\$\$SUB file is created

on the currently logged-in disk. Therefore, you cannot run a SUBMIT job from any other drive than A. (This occurs because the CCP will only search drive A for the \$\$\$SUB file.) After you make the changes shown in Listing One (page 94), the system will always create the \$\$\$SUB file on drive A.

Patch Procedure: Make a back-up copy of SUBMIT.COM before you use DDT to make the changes in Listing One.

SKIPNIF — Last But Not the Least

Last month David Kirkland shared with us his method of speeding up SUBMIT file truncation. When he submitted those valuable comments he also included his version of the original SKIPNIF program written by John Ramsey. Dave's program, SKIPNIF, allows one extra parameter on

the command line to specify how many records from the SUBMIT file are to be skipped. This is a valuable addition to an already excellent set of SUBMIT file control routines. I think this brings the subject of SUBMIT file processing to a logical conclusion. By using the many routines that have been published in DDJ over the past year, you can achieve a substantial level of control over how SUBMIT files are processed. A complete listing of Dave's SKIPNIF program can be found in Listing Two (page 94).

DDJ

(Listings begin on page 94)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 203.

CP/M users . . .

Make your move to 16-bit microcomputing... without ever leaving the 8-bit world.

From CP/M* to CP/M-86* or MS-DOS* and back...on command. That's the versatility CO-POWER-88™ brings to your CP/M machine.

CO-POWER-88 adds the power of 16-bit microcomputing—the potential for more memory, concurrent processing, multiaccess file systems and shared code—without impairing CP/M operation. Not only does your CP/M computer remain intact, you can even use CO-POWER-88 RAM as a simulated high-speed CP/M disk drive.

Built around the 8088 microprocessor, CO-POWER-88 provides dual-process-

ing power for almost any Z-80* or 8080 machine operating under CP/M (2.2)... including the Kaypro, Xerox, Big Board and ATR8000-CP/M personal computers.

A CO-POWER-88 costs only \$499.95 with 128 Kbytes of RAM, CP/M-86 with SWP's special BIOS is \$249.95 and 128-Kbyte RAM expansion (for 256 Kbytes total) is \$349.95. Or, save almost \$100 by ordering CO-POWER-88 already equipped with RAM expansion and CP/M-86—only \$999.95. Call for prices on CO-POWER-88/MS-DOS systems.

Available soon... IBM PC emulation software!

XEROX 820 Dual Density Upgrade	\$199.95
XEROX 820-II Ext. Dual Density Upgrade	\$100.00
BIG BOARD Dual Density Upgrade	\$150.00

SWP
MICROCOMPUTER PRODUCTS, INC.

We used to be Software Publishers, Inc.
2500 E. Randol Mill Road — 125
Arlington, Texas 76011
817/469-1181

ATR8000 and CO-POWER-88 are trademarks of SWP Microcomputer Products, Inc. CP/M and CP/M-86 are trademarks of Digital Research Corp. Kaypro is a trademark of Kaypro Non-Linear Systems, Inc. MS-DOS is a trademark of Microsoft Corp. Xerox 820 and Xerox 820-II are trademarks of Xerox Corp. Z-80 is a trademark of Zilog Corp.

Yes...I'd like to know more about SWP's ■ CO-POWER-88 ■ Dual Density Upgrades. Please rush me free literature.

Send to: SWP Microcomputer Products, Inc., Dept. 310
2500 E. Randol Mill Road (125)
Arlington, TX 76011

name

address

city

MAIL TODAY!

state zip

DEALER INQUIRIES
INVITED

PRICES AND SPECIFICATIONS
SUBJECT TO CHANGE WITHOUT NOTICE

Circle no. 96 on reader service card.

CP/M Exchange (Text begins on page 92)

Listing One

```
A>DDT SUBMIT.COM
DDT V2.2
NEXT PC
0600 0100
-D5BB
05BB 00 24 24 24 20 .$$$
05C0 20 20 20 20 53 55 42 00 00 00 1A 1A 1A 1A 1A 1A SUB...
05D0 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A .....
-
-S5BB
05BB 00 1
05BC 24 .
-G0
A>SAVE 5 SUBMIT.COM
```

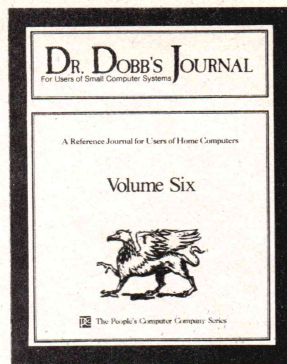
End Listing One

Listing Two

```
;
; SKIPNIF -- conditionally skip n commands within a submit file
;
; written by David Kirkland 6/12/83
; based on Don Wright's QUITIF (DDJ No. 71, Sept. 1982 at 8, 14).
;
; revised 6/31/83 to incorporate the "zero" option, based on
; John C. Ramsey's SKIPIF (DDJ No. 82, Aug. 1983 at 100-07).
;
; Usage:
;
; SKIPNIF Ambig fn n skip if fn is ambiguous
; SKIPNIF Exists fn n skip if fn exists
; SKIPNIF Missing fn n skip if fn does not exist
; SKIPNIF Null [ fn ] n skip if fn not specified not exist
; SKIPNIF Zero fn n skip if fn has zero length or does not exist
;
; Only the first character of the first keyword (the AMBIG or EXISTS or
; whatever) is checked. If an invalid keyword is entered, the
; n commands will be unconditionally skipped.
;
; This code written for Ithaca Intersystems IASM assembler; note
; that "ccir" is what most people call "cpir", and "ccdr" is "cpdr".
;
tpa equ 0100h
bdos equ 0005h
fcb1 equ 005ch ; first fcb prepared by CCP
name1 equ fcb1+1 ; start of filename of fcb1
fcb2 equ 006ch
name2 equ fcb2+1
cmdtail equ 0080h
;
drive equ 'M' ; drive on which $$$ SUB file resides
; ; ('A' for normal folk)
; BDOS codes
```

(Continued on page 96)

Volume 6 Now Available!



AT LAST!

ALL the Dr. Dobb's Journal issues from 1981, in one GIANT volume (562 pages). Send \$27.75 for volume 6, \$23.75 each for volumes 1-5, or \$125.00 for all 6 and SAVE!

Please add the following *per book*: \$2.50 for UPS, \$1.25 for U.S. Mail, or \$2.00 for Foreign postage. Delivery times are one week for UPS or 6-10 weeks for U.S. or Foreign Mail.

MAIL TO: Dr. Dobb's Journal
P.O. Box E Menlo Park, CA 94026

CP/M SYSTEMS COMPATABLE 8080/Z80 SOFTWARE

GRAFIX-PAC 1 ===== \$119 Object or \$250 Source ASM and VBASIC
(Vector Graphic/SuperBrain/IMSAI V10/SSM VB3)
BJACK Blackjack Standard Play against the House (Hawkeye)
BREAKO Breakout the Brick Wall with the Bouncing Ball
CHESS Graphics with Labeled Squares and Variable Level Play
CRAPS Full board Simulation with Help on Bet placement Odds
INVADERS Alien Invasion and YOU are Under Attack or Bunkers
OTHELLO Board game Plays Itself or You or You and Your Friend
POKER Poker Slot Style just Like the Machines at Los Vegas
POKERD Poker 5 Card Draw Against Hawkeye (watch the Bluffs)
STREK Startrek Missions into Uncharted Galaxies and Enemies
TARGETS Shooting Gallery full of Action with Detail Scoring

GRAFIX-PAC 2 ===== \$149 Object =====>Manual \$20
(Vector Graphic/SuperBrain/IMSAI V10/VB3/TRS M1+3)
VBASIC Graftix Language Interpreter with Full screen Editor
VRUN VBASIC Graftix Language Execute Only Compiler System
VDEMO Demonstration Program Source with Super Examples
VTEST Video Basic Functional Test Program Source code
CRUNCH Video Basic Program Optimizer to Conserve Memory

GRAFIX-PAC 3 ===== \$89 Object or \$239 Source VBASIC =====>Manual \$10
(Vector Graphic/SuperBrain/IMSAI V10/VB3/TRS M1+3)
SKETCH Advanced Designer Utility allows Creation of Screen
Graphics such as room layouts, schematics, logos, etc.
Screens can be Saved/reloaded and Printed on an EPSON
On line Help and Menu Selections for Drawing Functions
Circle/Block/Draw/Erase/Line/Clone/Lib/Label/Joystick

BIOS-PAC 1 ===== \$249 Source ZASM =====>Manual \$10
(Micropolis Systems (Vector Graphic/Sorcerer/Others)
BIOSDUAL Micropolis 5 inch Disk Driver with upto 25% Speed
improvement Including a Second 8 inch Disk Controller
Driver (C:+D:) >>Tarbell SD or DD, CCS, or AMD SBC.
Includes Drivers for Fast Mapped Video, AY3-8910 port
EPSON, TTY40, Sysgen/Format/Diskcopy and Submit Files
(requires MOVCPM and Vector or TDL Z80 Assembler)

*CP/M is a Registered Trademark of Digital Research



**HAWKEYE
GRAFIX**

Dial 213/348-7909 to Get Free Product Brochure

23914 MOBILE
CANOGA PARK
CA 91307 U.S.A.

Circle no. 47 on reader service card.

A Altos/ Apple II
with Hayes Micromodem II/Avatar/
Big Board/California Computer Sys-
tems 2710 and 2719/Casio/Cromemco/
CompuPro/DEC VT-180/ Eagle II/
Eagle III/Eagle IV/Escort/Exxon 500/
Godbout System Support Board/
Godbout Interfacer 3/Interfacer 4/Hayes
Micro-modem 100 or 80-103A/Imesai
SI02-2/Intertec/Ithaca/Kay Pro II/Lobo
Max-80/Monroe OC 8820/Morrow De-
cision 1/Morrow Micro Decision/
Northstar Advantage/Northstar
Horizon/Osborne I/Otrona/PMMI
MM-103A/Radio Shack Model II,
Model 12/Sanyo MBC-1000/Sanyo
MBC-1250/Sanyo MBC-2000/Sierra
Data Sciences ZSI0/Teletex/Tele-
video/ Vector 3/
Vector 4/ Xerox 820/
Xerox 820-II/Zenith 89/Zenith 90/Zenith 100/Zenith 110/
Zenith 120/ Zorba and more.

MITE

The most flexible
Data Communications Software
you can buy.
\$150.00

Dealer and distributor inquiries welcome.

**MYCROFT
LABS INC**

P.O. Box 6045, Tallahassee, FL 32314 (904) 385-1141

Circle no. 71 on reader service card.

CP/M Exchange (Listing continued, text begins on page 92)

Listing Two

```
;
PrtStr equ 9
Open equ 15
Close equ 16
Srchlst equ 17
Erase equ 19
Read equ 20
;
skipif org tpa
;
; first parse command line by
; - determining NSkip, the number of lines to skip if condition satisfied
; - ensuring that the contents of fcb2 at 6c contain the fn argument and
; not the NSkip argument (this problem occurs when the command is
; skipnif null NSkip
; i.e., when the fn argument actually is not entered and the skip should
; be executed).
;
lxi h,cmdtail
mov c,m ; length of command line
mvi b,0
dad b ; hl points to last character in command line
inx h
mvi m,' ' ; mark end of command line with space
; skip over "NSkip", the last argument on command line
call scanbl ; scan off trailing blanks
```



COMMUNICATIONS SOFTWARE

**WHY BUY 3 PROGRAMS?
LYNC HAS ALL 3 IN ONE!**

- 1) FILE TRANSFER 2) REMOTE OPERATION
- 3) COMMUNICATE WITH TIME-SHARE

*Easy set up on most systems.
Menu driven install program.
Command Mode of operation
Help Screen for Lync & Term*

Z80 based CP/M \$155
PC DOS \$155 TRSDOS \$75

We're Not the Biggest, Just the Best.

International Software Alliance (805) 966-3077
1835 Mission Ridge Santa Barbara, CA 93103

* DEALER AND OEM INQUIRIES INVITED *

FORMATS: 8" Standard; IBM PC DOS; TRS-80 Mod II with CP/M; VectorGraphic;
Cromemco; North Star; Osborne; Xerox 820 (8" or 5"); SuperBrain; NLS KayPro;
TeleVideo; TRS-80 Model I or III; Zenith/Heathkit; Victor 9000; Sanyo; Altos;
Fujitsu; Otrona.

CP/M is a trademark of DIGITAL RESEARCH. LYNC is a trademark of MIDNIGHT SOFTWARE.

Circle no. 51 on reader service card.

Q/C Version 3.1

Vacation? What's a vacation?

Jim Colvin just won't quit! He keeps improving his compiler for the popular C programming language. After releasing Version 3 in May, we told him to take a vacation. He did, for a whole week! But then he went to work and added **typedef**, **sizeof**, type casts, and function typing in Q/C Version 3.1. He also wrote six new i/o functions, including *setbuff()* and *setbsize()* to support large file buffers.

For only \$95, you get a ready-to-use C compiler for CP/M, as well as the *complete source code* for the entire package. This fully-supported compiler supports a large subset of C, and is upward compatible with the UNIX Version 7 compiler from Bell Labs. The Q/C library includes over 60 input/output and other support functions. Q/C features include:

- Complete source code for compiler and i/o library.
- No license fees on object code.
- 8080 and Z80 (with Zilog mnemonics) versions available.
- Produces ROMable code.
- Excellent support for assembly language.

In the 136-page *Q/C User's Manual* Jim tells you how to use the compiler and clearly describes each library function. There's even a chapter about the "internals" of the compiler. Jim puts his *home phone number* in the manual so you can get help evenings and weekends from the author, in addition to support during the day from The Code Works. (You can buy the manual for \$20, which applies toward purchase of Q/C.)

Write or call for details of Q/C Version 3.1. Disk formats include 8-in. single density, KayPro, Osborne DD, Morrow, Televideo 802, and Apple CP/M. VISA and MasterCard welcome.

**THE CODE
WORKS**

5266 Hollister
Suite 224
Santa Barbara, CA 93111
(805) 683-1585

Q/C, CP/M, Z80, and UNIX are trademarks of Quality Computer Systems, Digital Research, Zilog, Inc., and Bell Laboratories respectively.

Circle no. 18 on reader service card.


```

ccdr          ; scan backwards until a space reached
push         h          ; save for later

; now skip back over the next to last arg, which is "fn" if present
call        scanbl      ; find end of next arg
ccdr          ; find start of the arg

; now move back once more. Normally, there will be one argument left, the
; AMBIG or NULL or whatever. But in our special case of NULL with no
; fn entered, there will be nothing left.
call        scanbl      ; throw away extra blanks
mov         a,b          ; are there any characters unprocessed in the
ora         c            ; command line? if not, then there were only
jnz         DoNum        ; two arguments given on the command line

match        mvi         a, ' '          ; if only two args, add a blank for the second
sta         name2        ; arg to make fcb2 null
jmp         DoNum

; scanbl scans backwards over the command line discarding blanks;
; when a nonblank character is encountered, it stops.
; h -> str, bc = length remaining; return with hl -> first nonblank

scanbl       mov         a,b              ; see if any characters remain
ora         c
rz          ; none left
mvi         a, ' '
scanlp       cmp         m                ; *hl : ' '
rnz
dcx         h                            ; next char
dcx         b                            ; count down

```

(Continued on next page)



...nothing on Earth compares with DATAFLEX 2.0

Incredible power, un-precedented speed, and application flexibility for virtually any mission. That's DataFlex!

And now, the new DataFlex command language is the friendliest way you can deliver your software payload. All of the extraordinary state-of-the-art features of DataFlex 2.0 make it the clear choice for an applications data base.

But there's an even better reason to buy it... it grows with you. Needs change, your single user system may expand to multi-user, new requirements need to use valuable information developed in present applications. DataFlex will never keep you down!

DataFlex... available in single user and multi-user versions for 8 and 16 bit systems with extensive support for local area networks.

Get into orbit now with DataFlex.
The applications environment of the '80's.

DATAACCESS
CORPORATION

8525 S.W. 129 Terrace Miami, FL 33156
(305) 238-0012 TELEX 46-9021 Data Access CI

See us at
COMDEX
Fall '83

CP/M Exchange (Listing continued, text begins on page 92)

Listing Two

```

        jmp      scanbl

;
; PROCESS DECIMAL NUMBER--modified from ZCPR
;
DoNum
        pop      h
        inx      h
        inx      h          ; hl points to first char in number
; convert the decimal number pointed to by hl into binary and store
; at N2Skip
        mvi      c,0          ;C=ACCUMULATED VALUE set to zero
NUM1:
        MOV      A,M          ;GET CHAR
        CPI      ' '          ;DONE IF ' ', which marks end of number
        JZ       NUM2
        INX      H          ;PT TO NEXT CHAR
        SUI      '0'          ;CONVERT TO BINARY (ASCII 0-9 TO BINARY)
        CPI      10          ;ERROR IF >= 10
        JNC      NUMERR
        MOV      D,A          ;DIGIT IN D
        MOV      A,C          ;NEW VALUE = OLD VALUE * 10
        RLC
        RLC

        RLC
        ADD      C          ;CHECK FOR RANGE ERROR
        JC       NUMERR
        ADD      C          ;CHECK FOR RANGE ERROR
        JC       NUMERR
        ADD      D          ;NEW VALUE = OLD VALUE * 10 + DIGIT
        JC       NUMERR
        MOV      C,A          ;SET NEW VALUE
        jmp      NUM1          ;do next digit

;
;
NUMERR:
        mvi      c,PrtStr          ; come here if number invalid
        lxi      d,NErrMsg
        call     bdos
        ret

;
; come here once number finished
NUM2:
        MOV      A,C          ;GET ACCUMULATED VALUE
        sta      N2Skip          ; store the value for later

; now actually do what the caller asked for:

        lda      name1          ; first character of the AMBIG or EXISTS or ...
        lxi      h,name2

        cpi      'A'
        jz       ambig
        cpi      'E'
        jz       exists
        cpi      'M'
        jz       missing
        cpi      'N'
        jz       null

```

(Continued on page 100)

Micro Applications & Hardware

New Product—CP/M Plus (for Cromemco)

- Fast and more efficient operation.
- Hard Disk Support

- Bank Select available.
- Support for 1-16 drives of up to 512 megabytes each.

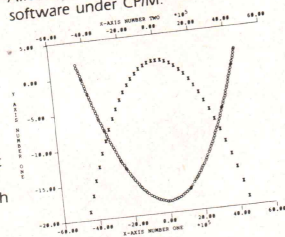
MICAH Software solutions for Cromemco

- CP/M 2.2™ industry standard implementation for Cromemco.
- MP/M™ multi-user, multi-tasking operating system.

- Expand™ CDS emulator... Allows you to run Cromemco software under CP/M.

Micro Plot

- Turn your Daisy Wheel or Thimble printer into an incremental plotter.
- Microsoft compatible subroutine library.
- Unique ability to plot and print text simultaneously.
- Prints horizontally, vertically or both at the same time.
- (actual reproduction reduced at right)



Select™ Word Processor

- Four Powerful Programs in one System \$299.
- Editor—Single key commands.
- SuperSpell—Automatic spelling correction.
- Teach—On screen tutorial.
- MergePrint—Personalized form letters done easily.

Epsys C Compiler (IBM PC, CP/M 86, MS DOS—\$250.)

- Floating point library for 8087.
- Librarian and linking loader.
- Debugger.
- Documentation.

MICAH

P.O. Box 896/Chesterfield/MO/63107 (314) 434-1896

Circle no. 65 on reader service card.

8748

CP/M SIMULATOR/DEBUGGER FOR THE INTEL 8748/8048

ANNOUNCING SIM48

- SIM48 allows you to load, trace, execute and save Intel 8748/8048 software using standard Intel Hex files.
- SIM48 allows you to simulate all instruction operations, timer/counter operations, I/O operations, interrupt processing, reset execution, internal and external RAM and ROM.
- SIM48's command set includes load, breakpoint, assemble, list (disassemble), trace, call and execute commands (as seen in DDT and ZSID).
- SIM48 allows you to simulate all software operations of the 8748/8048, yet costs 1/20th of an In-Circuit Emulator.
- SIM48 is CP/M compatible. Supplied on an 8" SSSD diskette.
- SIM22 (for Intel 8021/8022's) and SIM51 (for Intel 8751/8051's) soon to be released.

SIM48 \$150.00 SIM48 Manual \$20.00

Plus shipping and handling.
N.Y. State residents add sales tax.
Mastercard/Visa

Logical Systems

6184 TEALL STATION
SYRACUSE, NY 13217
(315) 457-9416

SIM48, SIM22, and SIM51 are trademarks of Logical Systems Corporation.
CP/M, ZSID, and DDT are trademarks of Digital Research.

Circle no. 59 on reader service card.

LYNX

the professional's replacement
for Microsoft L80

lynx



LYNX™ DOES

EVERYTHING L80™ DOES AND MORE!

LYNX™ LETS YOU USE ALL AVAILABLE MEMORY.

You can create COM files that are 10K larger—without overlays—by replacing L80 with LYNX.

LYNX™ IS A FULL OVERLAY LINKER

Running twice as fast as its nearest competitor, LYNX is tree structured, multi-segmented and multi-leveled, with automatic or explicit overlay invocation. You can run programs that are larger than available memory.

LYNX™ HAS BEEN HELPING MICROSOFT FORTRAN PROGRAMMERS FOR YEARS—NOW IT IS ALSO AVAILABLE FOR MICROSOFT BASIC AND AZTEC.™

LYNX™ IS A QUALITY PRODUCT from the same company who offers you:

- GrafTalk, the business graphics package for Micros
- GRAPHICS development tools
- 2780/3780, 3270, X.25 communications
- MICRO TO MICRO communications

 **REDDING GROUP INC.**

2730 High Ridge Road
Stamford, CT 06903
(203) 329-8874/Telex. 643351

LYNX and GrafTalk are trademarks of Redding Group, Inc. L80 is a trademark of Microsoft. AZTEC C is a trademark of MANX Software Systems.

Circle no. 88 on reader service card.

CP/M Exchange (Listing continued, text begins on page 92)

Listing Two

```
        cpi      'Z'
        jz       zero
                                ; if none of the above, fall through and skip

; skip over next N2Skip commands in $$$SUB
skip    mvi      c,Open          ; open $$$SUB
        lxi      d,subfile
        call     bdos
        inr      a                ; check if $$$SUB exists
        rz              ; if not, return

        lxi      h,subfile+14
        mvi      m,0              ; zero the S2 byte of the fcb to force a write
                                ; of the modified fcb back to the directory

        lda      N2Skip
        mov      c,a
        inx      h
        mov      a,m              ; decrement the number of records in $$$SUB
        sub      c                ; by N2Skip
        jp       notneg
        xra      a                ; if skip by more than in file, use zero
notneg   mov      m,a

        mvi      c,Close         ; close $$$SUB to rewrite the modified fcb
        lxi      d,subfile
        call     bdos

        mvi      c,PrtStr        ; tell the user what we've done
        lxi      d,skipmsg
        call     bdos

        xra      a                ; zero a and set flags for zero/missing
        ret

; abort if 2d parm ambiguous
ambig    mvi      a,'?'
        lxi      b,11
        ccir      ; cpir for people with normal assemblers
        jz       skip
        ret

; abort if file exists
exists   call     search
        jnz      skip
        ret

; abort if file missing
missing  call     search
        rnz
        jmp      skip

; abort if 2d parm null
null!    mvi      a,' '
        cmp      m
        rnz
        lxi      h,name2+8
        cmp      m
        rnz
        lda      fcb2
        ora      a
```



```

rnz                                ; d: given without more
jmp      skip

; abort if 2d parm is a zero length file
zero     call      missing          ; if file does not exist, its deemed zero length
rz       ; if so, missing has skipped N2Skip records in
; $$$ SUB, so now we return to the CCP

mvi      c,Open
lxi      d,fcB2
call     bdos                      ; open the specified file

mvi      c,Read
lxi      d,fcB2
call     bdos                      ; try to read the first record.
ora      a
rz       ; read was OK, so not zero length, don't skip.
jmp      skip                      ; read failed, zero length.

; search for first to see if any file exists to match fcb
search   mvi      c,Srch1st
lxi      d,fcB2
call     bdos
inr      a                        ; adjust flags: Z (-) not found
ret

;
N2Skip   db        0

skipmsg  db        "Next commands skipped",13,10,'$'
NErrMsg  db        "Illegal number to skip",13,10,'$'
subfile  db        drive+1-'A',"$$$ SUB",0,0,0,0
;

```

End Listing

SAVE YOUR 8BIT SYSTEM JOIN THE 16/32 BIT REVOLUTION THROUGH EVOLUTION, FOR UNDER \$800

HAVE YOU BEEN LOOKING FOR A 16/32 BIT SYSTEM WHICH DOES NOT OBSOLETE YOUR CURRENT Z80 HARDWARE OR SOFTWARE?

WOULD YOU LIKE TO RUN CPM86™, CPM68K™, MS-DOS™, OR UNIX™ ON YOUR Z80 SYSTEM?

WOULD YOU LIKE TO RUN IBM™ PC APPLICATIONS ON YOUR Z80 SYSTEM?

HAVE YOU BEEN LOOKING FOR A COST EFFECTIVE 16/32 BIT PROCESSOR SOFTWARE DEVELOPMENT SYSTEM?

WOULD YOU LIKE TO OFFER YOUR 16/32 BIT APPLICATIONS FOR USE ON Z80 BASED SYSTEMS?

WOULD YOU LIKE TO ADD AN INTELLIGENT HIGH SPEED RAM DISK TO YOUR Z80 BASED SYSTEM?

WOULD YOU LIKE TO MARKET YOUR Z80 SYSTEM AS AN 8/16 BIT SYSTEM?

IF YOU ANSWERED YES TO ANY OF THE ABOVE YOU SHOULD CONSIDER THE HSC CO16 ATTACHED RESOURCE PROCESSOR.

CO16 WITH EITHER 80186 OR 68000 MICRO PROCESSOR AND UP TO 768K BYTES OF RANDOM ACCESS PARITY CHECK MEMORY MAY BE ATTACHED TO VIRTUALLY ANY Z80 BASED SYSTEM. CO16 IS HOUSED IN AN ATTRACTIVE DESK TOP CASE WHICH CONTAINS THE 16 BIT PROCESSOR, MEMORY, AND POWER SUPPLY. CO16 IS DELIVERED WITH A COMPLETE SET OF DEVELOPMENT TOOLS AND THE MINIX OPERATING SYSTEM. CPM86 AND CPM68K ARE AVAILABLE AS OPTIONS.

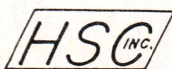
PRICES START AT \$660 FOR CO16 WITH 128K OF MEMORY, PRICE INCLUDES SOFTWARE AND DOCUMENTATION

HSC ALSO HAS COMPLETE CPM80 CROSS DEVELOPMENT SYSTEMS FOR DEVELOPING 68000, 8086/8088, 80186, AND 80286 SOFTWARE WHICH MAY BE PURCHASED SEPARATELY.

DEALER, DISTRIBUTOR, AND OEM DISCOUNTS ARE AVAILABLE

FOR INFORMATION CONTACT:

HSC INC.
BOX 86
HERKIMER, NEW YORK 13350
PHONE (315) 866 - 2311



Circle no. 44 on reader service card.

CAN YOU DO WITHOUT ... ??? ...

DISK MANAGER™ \$29.95*

DISK MANAGER™ is a utility program consisting of various functions needed for effectively using disks under CP/M®, especially by the Hard Disk User.

DISK MANAGER™ can run on any CP/M® system with any type of disks.

DISK MANAGER™ requires no installation.

These functions are included in DISK MANAGER™:

- Restore a deleted file.
- Establish multi-user links to a file.
- Change user number for a file.
- Map out bad blocks.
No more BDOS ERROR Bad Sector messages.
- Display the complete directory for the disk or a file.

DISK MANAGER™ is very easy to use. It is menu driven. It prompts for all parameters required by it to perform the selected function.

Now available for CP/M® 2.2

Contact Your Dealer or:



TRANTOR SYSTEMS, LTD.
4432 Enterprise Street, Unit I
Fremont, California 94538
(415) 490-3441
Telex: 17-1618 Attn: TNT

* Plus \$3.50 Shipping and Handling
Calif. Residents Add 6 1/2% Sales Tax

CP/M is a registered trademark of Digital Research

Circle no. 112 on reader service card.

16-BIT SOFTWARE TOOLBOX

by Ray Duncan

Version 2.0 of MS-DOS was enhanced with some 40 new function calls for use by application programs. These provide support for the hierarchical directories, device-independent I/O, and memory allocation. Most of them are rather radical departures from the CP/M-ish origins of MS-DOS and show a clear tendency toward providing a Unix-like environment.

In the process of preparing a PC-DOS programmer's reference book for publication next year, I have been trying each of the new functions out in turn by writing small assembly language utilities. When I reached function 4BH, called EXEC, I was especially intrigued. EXEC is designed to allow an application program (which we may call the "parent") to load and execute any other program from the disk, then regain control when the "child" is finished. The parent can pass a considerable amount of information to the child in the form of a command line, default File Control Blocks, and a set of strings called the "environment." Similarly, the child can pass an exit code back to the parent when it finishes its work.

The child program can in turn load other programs, and so on through many levels of control until you run out of memory. This function has plenty of uses even in a single-tasking, single-user operating system. For example, you could write your own screen-oriented, menu-driven command processor which would become the user's interface — one would never need to learn the more complex commands like MODE, CHDIR, and PATH (the same thing could be accomplished with a batch file but it would run much slower). Alternatively, you could load a second copy of the native Command Line Processor (COMMAND.COM), and let it run under the control of your program, rather than directly on top of MS-DOS.

In a multi-tasking environment such as we expect to see in MS-DOS 3.0, this function will be even more useful and will undoubtedly be elaborated with several additional features. Under such an operating system, a parent task can "spawn" any number of child tasks, which can execute concurrently and asynchronously, and communicate by means of queues, semaphores, and pipes.

Well, you say, pie in the sky is all very nice, but why is the EXEC function getting so much attention in this magazine column? The answer is, of course,

that when I tried to actually use the function I ran into any number of glitches and hazy spots in the documentation.

The "official" definition of the EXEC function, slightly boiled down, is given in Figure 1 (below). It looks straightforward enough, if you ignore the problem of what format the parameter block's DWORD addresses might be in. The documentation notes that "when your program received control, all available memory was allocated to it. You must free some memory (see call 4AH) before EXEC can load the program you are invoking. Normally, you would shrink down to the minimum amount of memory you need, and free the rest."

The manual goes on to state that all open files of the parent are duplicated in the newly created child task. Also inherited from the parent is some new creature called an "environment block," which is rather vaguely defined as a series of ASCII strings in the form "NAME=parameter."

Each of the strings in the environment block is terminated by a zero byte, and the whole set of strings is terminated by an additional zero byte. The environment always starts on a paragraph boundary, and its segment address is ultimately placed at offset 2CH of the Program Segment Prefix. You can either leave that slot in your EXEC parameter block as zero (in which case the child will acquire the

Call with:

AH = 4BH
AL = load type (see below)
DS = segment of ASCIIZ path & program name string
DX = offset of ASCIIZ path & program name string
ES = segment of parameter block
BX = offset of parameter block

Returns:

Carry Flag set if unsuccessful call. AL = error code.

Carry Flag cleared if successful call. Except for CS and IP, all other registers, including the stack pointers, are destroyed.

Load type = 0 : Load and execute program. "A program segment prefix is established for the program and the terminate and control/Break addresses are set to the instruction after the EXEC call."

Parameter block format:

WORD segment address of environment block
DWORD pointer to command line to be placed at new Program Segment Prefix + 80H
DWORD pointer to default FCB #1, to be passed at new Program Segment Prefix + 5CH
DWORD pointer to default FCB #2, to be passed at new Program Segment Prefix + 6 CH

Load type = 3 : Load overlay. Does not create a Program Segment Prefix, and does not begin execution of the overlay.

Parameter block format:

WORD segment address where file will be loaded
WORD relocation factor to be applied to the image

Figure 1.

Specification of arguments and results of MS-DOS EXEC function 4BH.

same environment block as your calling program), or you can provide a pointer to your own set of strings. The manual nonchalantly adds that the maximum size of the environment block is 32 Kbytes.

This zinger introduces a whole new set of mysteries. What in the world could the environment block possibly contain that could be as large as 32 Kbytes? Furthermore, the manual explicitly states that the environment block for any given program is static, implying that if more than one generation of child programs is resident in RAM, each one might have a distinct and separate copy of these potentially 32 Kbyte rascals floating around somewhere. My goodness!

Paging through the PC-DOS manual in search of further enlightenment, I found some help in appendix E which stated: "the environment built by the command processor (and passed to all programs it invokes) will contain a COMSPEC string at a minimum (the parameter on COMSPEC is the path used by DOS to locate COMMAND.COM on disk). The last PATH and PROMPT commands will also be in the environment, along with any environment strings entered through the SET command."

Thus, the environment block contains at most three strings that actually mean something to MS-DOS (PATH=, COMSPEC=, and PROMPT=). Any other strings found in the block by an application program were placed there for its information by a SET command issued by the operator or by another application program and have no effect on the behavior of the operating system proper. Loading a dummy COM file under the control of DEBUG, getting the pointer from offset 2CH in the Program Segment Prefix, and dumping out memory, I found that the "default" environment in my system is indeed only the two strings "PATH=" and "COMSPEC=A:\COMMAND.COM." Not so formidable after all.

The major remaining puzzle turned out to be the SETBLOCK function (MS-DOS function 4AH), which your program is supposed to use to "shrink down" the amount of space allocated to it before using EXEC to load another program. The manual says, "on entry, ES contains the segment of the block, BX contains the new requested block size in paragraphs." But when the "block" refers to the application program itself, and not to some piece of RAM it requested later, what segment to use: the segment of the Program Segment Prefix or the base of the Code Segment? Or does "block" actually refer to some kind of memory descriptor parameter block buried in COMMAND.COM?

Attempts to test this out with the Debugger were completely baffling, every attempt resulting in an error code. It turns out that since the Debugger essentially "owns" all of memory and loads your

HEATH...OSBORNE...KAYPRO...ANY CP/M*-80

Tired of 16-bit Snobbery?

C/NIX-80™

Brings

UNIX* Power to CP/M*-80

Features

Hierarchical Directories
UNIX*-like Shell
I/O Redirection & Pipes
On-Line Documentation
GREP, CAT, LS, CHDIR, etc.

Prices

C/NIX™-80 65\$
Manual Only 20\$

Specify Disk Format
Requires 48K CP/M-80

C/Craft™, Attn: T. Taft

22 Downing Road, Lexington, MA 02173

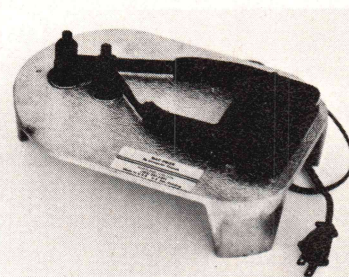
(617) 862-8177

*UNIX is a trademark of Bell Laboratories
CP/M is a trademark of Digital Research

Circle no. 16 on reader service card.

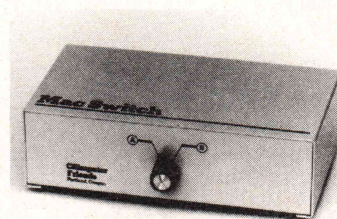
Re-ink any fabric ribbon for less than 5¢. Extremely simple operation. We have a **MAC INKER** for any printer. Lubricant ink safe for dot matrix printheads. Multi-colored inks, uninked cartridges available. Ask for brochure. Thousands of satisfied customers.

\$54⁹⁵ +



Mac Switch lets you share your computer with any two peripherals (serial or parallel). Ideal for word processors—never type an address twice. Ask us for brochure with tips on how to share two peripherals with **MAC SWITCH**. Total satisfaction or full refund.

\$99⁰⁰



Computer Friends

100 N.W. 86th Ave.
Portland, OR 97229
503/297-2321

**Mac Inker
& MacSwitch**

Circle no. 22 on reader service card.

program via its own mechanisms, a SET-BLOCK call by a program running under the control of DEBUG is illegal by definition — from the operating system's point of view, it is trying to modify a memory allocation that does not "belong" to it. In any event, the "segment of the block" that should be fed to function 4AH is the Program Segment Prefix.

Finding the number of paragraphs to "shrink" your program's memory allocation down to is also a little tricky, at least for EXE files. You might think that you can simply make an Equate to an address called Program_End, put that in a data segment at the end of your program, and load that segment address into BX at execution time with the aid of the GROUP modifier, thereby letting the Linker and Loader do all the work.

Unfortunately, that method won't work — at least not all the time. The Linker moves all the declared segments around to suit its own purposes, and the

physical order of segments in your source code doesn't usually correspond to the order of segments in the resultant EXE file. The best way I have figured out to get the total size of the program at run-time is to put an equate to "Program_End" into another module, which is separately assembled, and reference it with an EXTRN in your main program. At link time, specify the module containing the "Program_End" equate as the last in the list of OBJ files.

Of course, you can also "cheat" by simply assembling and linking the program, looking at the size of the EXE file in the directory listing, and then reassembling the program with the correct size in the code as a literal value. This assumes that all data areas used by the program are embedded as declared constants and variables. An approach like this could cause you trouble later if you modify the program in a way that increases its size, then forget to update the literal too.

To keep the amount of painful interaction between the EXEC function and DDJ readers to a minimum, I am publishing herewith a very detailed example program named FXN4BH (see the listing, below), using EXEC to load the MS-DOS utility CHKDSK.COM, that is guaranteed to work. It requires that both FXN4BH.EXE and CHKDSK.COM be present on the "current" or default disk drive. FXN4BH can fail if there is not enough room above it in RAM to load and execute CHKDSK.COM, or if the transient portion of COMMAND.COM in highest RAM (which contains the actual loader) has been destroyed and there is not enough free memory to reload it.

DDJ

(Listing begins below)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 205.

16-Bit Toolbox Listing (Text begins on page 102)

```

1      name      FXN4BH
2      page      55,132
3      title     'FXN4BH --- demo PC-DOS EXEC function'
4
5      ;
6      ; FXN4BH --- demonstrate use of the
7      ; PC-DOS 2.0 EXEC function call 4BH
8      ;
9      ; Copyright (c) 1983 by Ray Duncan
10     ;
11     = 000D      cr      equ      0dh      ;ASCII carriage return
12     = 000A      lf      equ      0ah      ;ASCII line feed
13     0000      cseg     segment para public 'CODE'
14
15               assume   cs:cseg,ds:data,ss:stack
16
17     0000      demo     proc      far
18
19               ;at entry DS & ES = PSP
20     0000 1E      push    ds          ;Save address for final
21     0001 33 C0    xor     ax,ax      ;FAR RET to PC-DOS on stack
22     0003 50      push    ax
23
24               ;save copy of SS:SP for use
25               ;after return from overlay
26     0004 2E: 8C 16 0050 R    mov     cs:STK_SEG,ss
27     0009 2E: 89 26 0052 R    mov     cs:STK_PTR,sp
28
29               ;
30               ;Reserve 1000H bytes for
31               ;this loader and release
32               ;the rest of memory for
33               ;use by the overlayed program.
34     000E BB 0100    mov     bx,100h  ;ES=segment of PSP of loader

```

(Continued on page 106)

ALTECON RESEARCH GROUP



MERGE SORT

- Written in C for portability
- Includes complete source code
- Documentation on source code
- Disk based, no file size restrictions
- Sorts fixed and variable length records
- ASCII and dictionary order sort modes
- Reverse order sorting mode
- Compatible with 8-bit CP/M systems
- Only \$29.95 on 8-inch SSSD disk (Other formats inquire)

For further information on Merge Sort
Call or Write

Altecon Research Group
Post Office Box 11390
Glendale, CA 91206 U.S.A.
Telephone: (213) 240-6884

CP/M is a trademark of Digital Research, Inc.

Circle no. 4 on reader service card.

PORTABLE 68000 ASSEMBLER

MOTOROLA SYNTAX
LINKER
OBJECT LIBRARIAN
CROSS REFERENCE
STRUCTURED PROGRAMMING FEATURES

OBJECT CODE OPTIMIZED
FOR CP/M-80, CP/M-68K
OR IBM-PC

\$300

PORTABLE C SOURCE

\$750

Quelo
843 NW 54th
Seattle, WA 98107

(206) 784-8018
mornings
Dick Curtiss

CP/M is a trademark of Digital Research
IBM-PC is a trademark of IBM

Circle no. 84 on reader service card.

PROGRAMMERS-

COMPARE Products. Get Answers.

We research and specialize only in software for programmers of micros. If a product is unknown to you and we recommend it, then we take the risk.

PHILOSOPHY: We carry every programmer's product for CPM80, MSDOS and CPM86 plus every key product for APPLE, Commodore 64, ATARI and TRSDOS.

"C" LANGUAGE		LIST PRICE	OUR PRICE
APPLE:AZTEC C-Full, ASM	8086	\$199	\$179
8080:BDS C-Fast, popular	8086	150	125
8080:AZTEC C-Full	8086	199	179
Z80: ECOSOFTE-Fast, Full	8086	350	315
8086:C86- optimizer, Meg	8086	399	call
Lattice - Full	8086	500	395
MicroSoft (Lattice)	MSDOS	500	395
Digital Research - Megabyte	8086	350	280
Desmet by CWare-Fast	8086	109	99
BASIC		ENVIRONMENT	LIST PRICE
MBASIC-80 - MicroSoft	8080	8080	\$375
BASCOM-86 - MicroSoft	8086	8086	395
CB-86 - DRI	CPM86/PC	600	399
PBASIC - DRI	8086	150	119
Business BASIC-MicroS	MSDOS	600	449
EDITORS		Programming	LIST PRICE
BELLESOF - PASCAL	MSDOS	8086	\$295
C Screen Editor source	8080/86	NA	60
EDIX	PCDOS	195	149
FINAL WORD - Manuals	8080/86	300	225
MINCE	CPM, MSDOS	175	149
PMATE - powerful	CPM	195	175
	8086	225	195
VEDIT - full, liked	CPM, PCDOS	150	119
	8086	200	159
PASCAL		ENVIRONMENT	LIST PRICE
PASCAL MT+86	CPM86	8086/86	\$400
without SPP	CPM80	350	259
MS PASCAL 86 -	MSDOS	350	280
ALCOR PASCAL full w/	8080	250	call
right extensions to learn			
SBB PASCAL great, fast	PCDOS	350	315
PASCAL 64 - nearly full	COM 64	99	89
UCSD PASCAL Compiler	UCSD-PC	375	call
LANGUAGE LIBRARIES		ENVIRONMENT	LIST PRICE
C-to-dBASE - interface	8086/86	NA	\$95
C TOOLS - Graph, Str...	PCDOS	NA	115
FLOAT87 - Lattice, PL1	PCDOS	NA	115
GRAPH-GSX-80	CPM80	60	50
HALO	PCDOS	150	125
GRAPHLIB	CPM80	400	365
ISAM: Access Manager-86	8086	400	300
PHACT - with C	PCDOS	250	225
FABS	CPM80	150	135
PASCAL TOOLS	PCDOS	NA	115
SCREEN: Display Mgr-86	8086	500	375
PANEL-86 - many	PCDOS	NA	315
FORTRAN		ENVIRONMENT	LIST PRICE
MS FORTRAN-86 - Megabyte	MSDOS	350	\$265
SS FORTRAN-86	CPM-86	425	345
FORTAN-80 - 66 decent	CPM80	500	350
INTEL FORTRAN-86	IBM PC	NA	1400
ASSEMBLER		ENVIRONMENT	LIST PRICE
MAC/65 - macros, fast	ATARI	80	\$80
MACRO-80 -	CPM80	200	149
ORCA/M - Reloc., Macro	APPLE	99	91
RMAC. etc. (Prog. Util)	CPM80	200	159
	8086	200	159
OPERATING SYSTEMS		ENVIRONMENT	LIST PRICE
CPM-86	PC	\$60	\$49
"CPM86 Emulator"	MSDOS	NA	95
MicroShell upgrade CPM	8080	150	125
"MSDOS Emulator"	CPM86	NA	95
"Multitasking PCDOS"	PCDOS	239	179
ONX - real time UNIX	PC	650	call
TOOLKITS		ENVIRONMENT	LIST PRICE
Carousel	CPM80	\$249	\$225
MicroTools	CPM80	150	125
NORTON for 1.1 or 2.0	PCDOS	80	65
NORTON Book with Disk	PCDOS	80	65
Power	CPM80/PC	169	139
Programmer's Toolkit	8080/86	NA	95
COBOL		ENVIRONMENT	LIST PRICE
CIS COBOL - no royalty	CPM80/86	\$800	\$600
Level II COBOL - High	8086	1600	call
MS COBOL-86	MSDOS	750	595
RM/COBOL-80	CPM80	750	595
RM/COBOL-86	8086	950	575
LOOKING FOR		ENVIRONMENT	LIST PRICE
Full and decent "C" compiler for	CPM-80	\$300	\$259
Commodore 64 and/or for Atari.	ADA - Janus with tools	500	449
PROLOG for MSDOS.	PCDOS	175	call
OTHER PRODUCTS		ENVIRONMENT	LIST PRICE
ADA - Janus	CPM-80	\$300	\$259
ADA - Janus with tools	PCDOS	500	449
LISP - IQ LISP - full,	PCDOS	175	call
MAGIC/L - Intriguing	CPM80	295	call
MODULA II - PCCode	8086	495	455
MODULA II - PCCode	APPLE	495	455
PL/1-86 - DRI	CPM-86	750	560
PI/1-80 - DRI	8080	500	369
ASCOM 80 CPM80	\$175	\$159	
CACHE/Q Virtual Mem.	CPM80/PC	225	185
COMM-PAC - 8086	200	179	
"Read CPM86" MSDOS	NA	95	
"Read MSDOS"	CPM86	NA	50

Call for a catalog, literature, comparisons, prices. Shipping \$2.50 per item purchased. All FORMATS available.

THE PROGRAMMER'S SHOP™

908-1 Providence Highway, Dedham, MA 02026,
617-461-0120, Mass: 800-442-8070

VISA 800-421-8006 MASTER CARD

RECENT DISCOVERIES HIGHLIGHTED

SBB PASCAL for PCDOS supports IBM and MS PASCAL with no changes. Consistently benchmarked faster. Includes runtime source to give you control. Guaranteed. \$315.

- SERVICES**
- * Programmer's Referral List
 - * Compare Products
 - * Help find a Publisher
 - * Evaluation Literature
 - * Bulletin Board
 - * Dealers Inquire
 - * Newsletter

Circle no. 80 on reader service card.

16-Bit Toolbox Listing (Listing continued, text begins on page 102)

```

32      0011 B4 4A      mov     ah,4ah    ;BX=paragraphs to reserve
33      0013 CD 21      int     21h
34
35                      ;make the messages in data
36                      ;segment addressable
37      0015 BB ---- R   mov     ax,seg DATA
38      0018 8E D8      mov     ds,ax
39      001A 8E C0      mov     es,ax
40
41                      ;jump if memory
42                      ;de-allocation failed
43      001C 72 2A      jc      ALLOC_ERR
44                      ;print memory successfully
45                      ;released
46      001E BA 001E R   mov     dx,offset MSG2
47      0021 B4 09      mov     ah,9
48      0023 CD 21      int     21h
49
50                      ;
51                      ;now load and execute
52                      ;the overlaid program.
53      0025 BA 00A4 R   mov     dx,offset PGM_NAME
54      0028 BB 00B0 R   mov     bx,offset PAR_BLK
55      002B B0 00      mov     al,0
56      002D B4 4B      mov     ah,4bh
57      002F CD 21      int     21h
58
59                      ;restore stack pointers
60                      ;to state before EXEC call
61      0031 2E: 8E 16 0050 R   mov     ss,cs:STK_SEG
62      0036 2E: 8B 26 0052 R   mov     sp,cs:STK_PTR
63
64                      ;Make data segment
65                      ;addressable again
66      003B BB ---- R   mov     ax,seg DATA
67      003E 8E D8      mov     ds,ax
68
69                      ;print message that loader
70                      ;successfully regained control
71      0040 BA 0059 R   mov     dx,offset MSG3
72      0043 B4 09      mov     ah,9
73      0045 CD 21      int     21h
74
75                      ;now exit to PC-DOS
76      0047 CB      ret
77
78      0048      alloc_err:      ;come here if memory
79                      ;cannot be released
80      004B BA 0000 R   mov     dx,offset MSG1
81      004D B4 09      mov     ah,9
82      004F CD 21      int     21h    ;print error message and
83      0050 CB      ret            ;exit to PC-DOS
84
85                      ;
86                      ;
87                      ;these two variables must
88                      ;reside in Code Segment so
89                      ;that they are addressable
90                      ;after return from overlay.

```

(Continued on page 108)

A Professional Quality Z80/8080 Disassembler

REVAS Version 3

Uses either ZILOG or 8080 mnemonics
Includes UNDOCUMENTED Z80 opcodes
Handles both BYTE (DB) & WORD (DW) data
Disassembles object code up to 64k long!
Lets you insert COMMENTS in the disassembly!

A powerful command set gives you:

INTERACTIVE disassembly
Command Strings & Macros
On-line Help
Calculations in ANY Number Base!
Flexible file and I/O control
All the functions of REVAS V2.5

REVAS:

Is fully supported with low cost user updates
Runs in a Z80 CPU under CP/M*
Is normally supplied on SSSD 8" diskette
Revas V 3...\$90.00 Manual only...\$15.00
California Residents add 6½% sales tax

REVASCO

6032 Charlton Ave., Los Angeles, CA. 90056
(213) 649-3575

*CP/M is a Trademark of Digital Research, Inc.

Circle no. 89 on reader service card.



CP/M (TM) Digital Research, Inc.

CDOS (TM) Cromemco, Inc.

RUN CDOS PROGRAMS UNDER CP/M 2.2 RUN CP/M 2.2 PROGRAMS UNDER CDOS

- Z80 code • Customizable system calls
- No program, CP/M, or CDOS modifications

INTRCEPT Version I-1 **\$89.95**

- emulates CDOS under CP/M 2.2

INTRCEPT Version I-2 **\$89.95**

- emulates CP/M 2.2 under CDOS

INTRCEPT Version II **\$129.95**

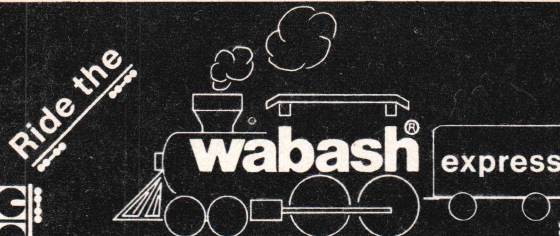
- emulates both CDOS and CP/M 2.2
- add \$30.00 for CDOS emulator source code

Check, VISA, MC • In CA, add 6% tax.

PRO microSystems

16609 Sagewood Lane
Poway, California 92064
(619) 578-1240

Circle no. 81 on reader service card.



5¼" \$170*

SINGLE SIDE
SINGLE DENSITY
W/HUB RING

100% CERTIFIED
2 YEAR WARRANTY

SOFT,
10 OR 16
SECTORS

8" \$199*

SINGLE SIDE
SINGLE DENSITY

100% CERTIFIED
2 YEAR WARRANTY

SOFT
OR 32
SECTORS

5¼" \$199*

SINGLE SIDE
DOUBLE DENSITY
W/HUB RING

100% CERTIFIED
2 YEAR WARRANTY

SOFT,
10 OR 16
SECTORS

8" \$249*

SINGLE SIDE
DOUBLE DENSITY

100% CERTIFIED
2 YEAR WARRANTY

SOFT
OR 32
SECTORS

5¼" \$299*

DOUBLE SIDE
DOUBLE DENSITY
W/HUB RING

100% CERTIFIED
2 YEAR WARRANTY

SOFT,
10 OR 16
SECTORS

8" \$309*

DOUBLE SIDE
DOUBLE DENSITY

100% CERTIFIED
2 YEAR WARRANTY

SOFT
OR 32
SECTORS

- * Minimum order 10
- * Packed 10 boxes of 10 diskettes with sleeves and labels
- * Quantity discounts - 100 deduct 5%,
1,000 deduct 7%, 5,000 deduct 10%
- * Add \$5.00 per case 5¼", \$7.00 per case 8" (case of 100)
For shipping and handling Continental U.S.A., U.P.S. ground.

VINYL STORAGE PAGES

5¼" or 8"

10/\$5

SNAP-IT POWER CENTER

- Turn one outlet into six
- Shock-safe
- Unbreakable
- 15 Amp Circuit Breaker
- Lighted On-Off Switch



\$19.95

HARDHOLE DISK PROTECTORS

Reinforcing rings
of tough mylar
protect disk hole
edge from damage.
Applicators \$3 \$4
Hardhole Rings (50) \$6 \$8



DISK DRIVE HEAD CLEANING KITS

Prevent head
crashes and
ensure error-free
operation
5¼" or 8" \$19.50



LIBRARY CASES

8" Kas-sette/10 \$2.99

5¼" Mini Kas-sette/10... \$2.49

SFD C-10 CASSETTES... 10/\$7

(All cassettes include box and labels.)

Get 8 cassettes, C-10
Sonic, and Cassette/8
Library-Album,
as illustrated,
for only \$8



We also stock at FANTASTIC low prices

MAXELL 3M DYSPAN

BASF OPUS

Floppies, Tape, Data Cartridges,
Data Cassettes, and Disk Packs

- Written purchase orders accepted from government agencies and well rated firms for net 30 day billing.
- International orders accepted with a 15.00 surcharge for handling, plus shipping charges.
- C.O.D. requires a 10% deposit.
- We accept Visa, Mastercard, Money Orders, and Certified checks.
- Checks require bank clearances.
- All shipments F.O.B. San Diego.
- Minimum shipping and handling 2.00, minimum order 10.00.
- California residents add 6% sales tax. Prices and terms subject to change without notice.
- All sales subject to availability, acceptance, and verification.
- All sales are final.
- Satisfaction guaranteed or full refund.

We also offer printer ribbons, printwheels, type elements, equipment covers, power consoles, paper supplies, storage and filing equipment, furniture and many other accessories for word and data processing systems. Write for our free catalog.

Orders Only
800-854-1555

Information
619-268-3537

Modem Hotline (Anytime)
619-268-4488

Exclusive Monthly Specials

ABC
DATA PRODUCTS
(FORMERLY ABM)

ITT TELEX 4992217
8868 CLAIREMONT MESA BLVD
SAN DIEGO, CALIFORNIA 92123

Circle no. 1 on reader service card.

16-Bit Toolbox Listing (Listing continued, text begins on page 102)

```

84      0050 0000          stk_seg dw      0          ;original SS contents
85      0052 0000          stk_ptr dw      0          ;original SP contents
86
87      0054              cseg      ends
88
89                                ;declare a stack area
90                                ;for use by this loader
91      0000          stack      segment      para stack 'STACK'
92                                ;allow 64 bytes in this case
93      0000      40 [      db      64 dup (?)
94                                ??
95                                ]
96
97      0040          stack      ends
98
99                                ;declare data segment to
100                                ;contain variables and tables
101      0000          data      segment      para public 'DATA'
102      ;
103      0000 0D 0A          msg1      db      cr,lf
104      0002 55 6E 61 62 6C 65      db      'Unable to release memory.'
105      20 74 6F 20 72 65

```

IBM PC-8087 SUPPORT FROM MICROWARE

87FORTRAN/RTOS™ is a MicroWare adaptation of the Intel Fortran-86 compiler, a full ANSI-77 subset with 8087 extensions and overlays. It generates in line 8087 code allowing use of all 8087 data types, including 80 bit reals and 64 bit integers. The complete subset I/O is supported including Internal and External Files and List Directed I/O. 87FORTRAN/RTOS uses the Intel large memory model, allowing data/code structures which utilize the full megabyte. The compiler provides direct access to 8088 ports and supports logical operations on 8 and 16 bit operands normally treated in assembly language. It is ideal for large applications which are number intensive or control hardware.

87PASCAL/RTOS™ is Intel's ISO-Standard Pascal with 8087-8088 exceptions. These make it possible to use all the 8087 data types directly, while generating modules in one of the three Intel Memory Models.

87FORTRAN/RTOS and

87PASCAL/RTOS include RTOS and support for one year. each **\$1350**

PC TECH JOURNAL REVIEW:

"The MicroWare package is preferable ... it executes the basic operations more rapidly and MicroWare provides a free update service."

87BASIC™ includes patches to the IBM Basic Compiler and both runtime libraries for USER TRANSPARENT and COMPLETE 8087 support. Provides super fast performance for all numeric operations including trigonometrics, transcendentals, addition, subtraction, multiplication, and division. **\$150**

87MACRO™ - our complete 8087 software development package. It contains a "Pre-processor" which converts 8087 instructions into 8088 escape codes, source code for a set of 8087 macros, and an object library of 8087 functions which include transcendentals, trigonometrics, hyperbolics, encoding, decoding and conversions. **\$150**

PLM-86 with RTOS. **\$1100**

MICROWARE BEST SELLERS!

8087-3 CHIP in stock with 180 day warranty and 8088 exchange **\$223**

FASTPAK™+8087 includes one runtime library (87BASIC or 87MACRO), the 8087 chip, the 87/88GUIDE, and installation instructions **\$375**

MATRIXPAK™ manages a **MEGABYTE!**

Written in assembly language, our runtime package accurately manipulates large matrices at very fast speeds. Includes matrix inversion and the solution of simultaneous linear equations. Callable from MS Fortran 3.13, MS Pascal 3.13, SSS Fortran, and MicroWare 87MACRO, 87BASIC, and RTOS compilers each **\$150**

87/88GUIDE - an excellent tutorial on writing 8087 code and interfacing it with compilers. Full of code that runs! **\$30**

MICROSOFT FORTRAN 3.13

MICROSOFT PASCAL 3.13 These new IEEE compatible compilers support both double precision and the 8087. each **\$289**

MICROSOFT C COMPILER

includes Lattice C and the MS Librarian. **\$399**

64K RAM Upgrade	59
SuperSoft Fortran	340
SSS 8087 Support	50
SuperSoft Voice Drive	895
87BASIC+	75
TRACE86 Utility	125
Microsoft Business Basic Compiler	495
Computer Innovations C86	345
STSC APL★PLUS/PC	545
HALO Graphics	call
Energraphics	call
Sandstar products	call
Novation 212 SmartCat	495

**Micro
Ware**

P.O. Box 79
Kingston, MA
02364
(617) 746-7341

**You Can
Talk To Us!**

Circle no. 69 on reader service card.


```

106      6C 65 61 73 65 20
107      6D 65 6D 6F 72 79
108      2E
109      001B 0D 0A 24      db      cr,lf,'$'
110      001E 0D 0A      msg2      db      cr,lf
111      0020 4D 65 6D 6F 72 79      db      'Memory above loader released.'
112      20 61 62 6F 76 65
113      20 6C 6F 61 64 65
114      72 20 72 65 6C 65
115      61 73 65 64 2E
116      003D 0D 0A 4E 6F 77 20      db      cr,lf,'Now loading CHKDSK.COM.'
117      6C 6F 61 64 69 6E
118      67 20 43 48 4B 44
119      53 4B 2E 43 4F 4D
120      2E
121      0056 0D 0A 24      db      cr,lf,'$'
122      0059 0D 0A      msg3      db      cr,lf
123      005B 4C 6F 61 64 65 72      db      'Loader regained control from CHKDSK,'
124      20 72 65 67 61 69
125      6E 65 64 20 63 6F
126      6E 74 72 6F 6C 20
127      66 72 6F 6D 20 43
128      48 4B 44 53 4B 2C
129      007F 0D 0A      db      cr,lf
130      0081 6E 6F 77 20 6D 61      db      'now making final exit to PC-DOS.'
131      6B 69 6E 67 20 66

```

(Continued on next page)

At Last! bds C... Ver. 1.5

Including a new dynamic debugger Still the choice of professionals

- Compiler option to generate special symbol table for new dynamic debugger by David Kirkland. (With the debugger, the distribution package now requires two disks.)
- Takes full advantage of CP/M* 2.x, including random-record read, seek relative to file end, user number prefixes, and better error reporting.
- Click option to suppress warm-boot
- New library file search capabilities
- New, fully-indexed 180 page manual
- * CP/M is a trademark of Digital Research, Inc.

V 1.5 \$120.00
V 1.46 \$115.00
(needs only 1.4 CP/M)

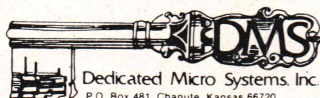
Other C compilers and
C related products
available... Call!

TERMS: CHECK,
MONEY ORDER, C.O.D.,
CHARGE CARD
HOURS: 9 am—5 pm
Monday—Friday
(316) 431-0018

IT'S HERE! MONEY MATH

- Uses BCD internal representation.
- You choose from two types of rounding.
- Configurable exception handling
- Distributed with 12 digits precision. Easily configured for more or less
- Excess 64 exponents

SOURCE INCLUDED \$5000



Dedicated Micro Systems, Inc.
P.O. Box 481, Chanute, Kansas 66720

FOR \$29.95, MICRO-WYL WILL WHEN OTHERS WON'T!

You'll know why INFOWORLD rated *Micro-WYL's* performance "Excellent" when you want to insert a file into a file; when you want to print directly from the screen; when you want to see what your text will look like before printing; set tabs where you want them; write programs the easy way; and do many other things that no other word processor can do. *Micro-WYL* is a line editor; use it in conjunction with EMACS derivative word processors and you have the best of all worlds! For \$29.95, plus \$2.00 postage and handling, have it your way! Sold with the usual Overbeek 30 day money-back guarantee.

LET *Micro-WYL* DO IT ALL TODAY!
Enclosed find check for \$31.95 or

charge my Mastercard# _____ Expires: _____

Visa# _____ Expires: _____

Check format desired:

<input type="checkbox"/> 8" SSSD	<input type="checkbox"/> Osborne Single Density	<input type="checkbox"/> KayPro II
<input type="checkbox"/> Superbrain	<input type="checkbox"/> Osborne Double Density	<input type="checkbox"/> NEC 5"
<input type="checkbox"/> Northstar Advantage	<input type="checkbox"/> Morrow Micro Decision	<input type="checkbox"/> Televideo802
<input type="checkbox"/> Northstar Horizon	<input type="checkbox"/> Xerox 820 Single Density	<input type="checkbox"/> ALTOS 5
<input type="checkbox"/> Apple/Softcard	<input type="checkbox"/> Xerox 820 Double Density	<input type="checkbox"/> Epson

I'm not ready to order now, but send me information about all the affordable programs from Overbeek Enterprises.

Name _____
Address _____
City _____ State _____ Zip _____

OVERBEEK ENTERPRISES, P.O. Box 726D, Elgin, IL 60120
312-697-8420

Micro-WYL—another affordable program from Overbeek Enterprises.

16-Bit Toolbox Listing (Listing continued, text begins on page 102)

```

132      69 6E 61 6C 20 65
133      78 69 74 20 74 6F
134      20 50 43 2D 44 4F
135      53 2E
136 00A1 0D 0A 24      db      cr,lf,'$'
137
138                                ;drive, path, and name of program
139                                ;to be loaded and executed.
140 00A4 5C 43 48 4B 44 53      pgm_name db      '\CHKDSK.COM',0
141      4B 2E 43 4F 4D 00
142
143 00B0 ---- R      par_blk dw      ENVIR      ;segment address of
144                                ;environment descriptor
145
146                                ;full address of command line
147                                ;to be passed at offset 80H
148 00B2 00BE R      dw      offset CMD_LINE      ;in overlaid
149 00B4 ---- R      dw      seg CMD_LINE      ;program's PSP
150
151                                ;full address of default
152                                ;File Control Block to be
153                                ;passed at offset 5CH in
154 00B6 00C5 R      dw      offset FCB1      ;overlaid
155 00B8 ---- R      dw      seg FCB1      ;program's PSP
156
157                                ;full address of default
158                                ;File Control Block to be
159                                ;passed at offset 6CH in
160 00BA 00EA R      dw      offset FCB2      ;overlaid
161 00BC ---- R      dw      seg FCB2      ;program's PSP
162
163                                ;actual command line tail
164                                ;to be passed to overlay
165 00BE 04 20 2A 2E 2A 0D      cmd_line db      4,' *.*',cr,0
166      00
167
168                                ;first default FCB to
169 00C5 00      fcb1      db      0      ;be passed to overlay
170 00C6      0B [      db      11 dup ('?')
171      3F
172      ]
173
174 00D1      19 [      db      25 dup (0)
175      00
176      ]
177
178
179                                ;second default FCB to
180 00EA 00      fcb2      db      0      ;be passed to overlay
181 00EB      0B [      db      11 dup (' ')
182      20
183      ]
184

```


185	00F6	19	[db	25 dup (0)
186			00		
187]		
188					
189				;	
190	010F			data	ends
191					
192					;declare separate data
193					;segment to contain
194					;environment descriptor
195	0000			envir	segment para 'ENVIR'
196					;
197					;Search path used by PC-DOS
198					;to look for commands or
199					;batch files not found in
200	0000	50	41 54 48 3D 00	db	'PATH=',0 ;the current directory
201					;
202					;Search path used by PC-DOS
203					;to locate COMMAND.COM
204	0006	43	4F 4D 53 50 45	db	'COMSPEC=A:\COMMAND.COM',0
205		43	3D 41 3A 5C 43		
206		4F	4D 4D 41 4E 44		
207		2E	43 4F 4D 00		
208	001D	00		db	0 ;extra 0 byte designates
209					;end of environment
210	001E			envir	ends
211					
212				end	demo

DEVELOPMENT HARDWARE/SOFTWARE
GTEK MODEL 7128 EPROM PROGRAMMER

TRANSFORM

Company: MasterComputing, Inc., P.O.
Box 17442, Greenville, SC 29606

Price: \$39.95

Circle Reader Service No. 121

Reviewed by Michael Favitta

MasterComputing's (MCI) TRANSFORM is a language preprocessor for Microsoft BASIC and other compatible BASIC's. It translates MCI's enhanced BASIC syntax (which includes block structured statements) into Microsoft BASIC 5.X syntax. The purpose of TRANSFORM is to allow BASIC programmers to develop programs that are easier to debug, understand, and maintain, by using a structured source language.

TRANSFORM is designed for CP/M systems that have 48K of memory. Translated programs will run under MBASIC 5.X or can be compiled with BASCOM 5.3. The translator is distributed on an 8-inch single-density disk along with a sample program and a utility program named SIFT.

A Structured BASIC Language

MCI has made three changes to Microsoft BASIC to create an enhanced version of BASIC. MCI has added blocked control structures, an INCLUDE facility, and replaced line numbers with labels.

To add blocked structures to BASIC, the Microsoft BASIC statements ON-GOTO and IF-THEN-ELSE were modified and a new statement, REPEAT-UNTIL, was added. The syntax of all other statements remains the same. The end-of-block sentinels, ENDGOTO and ENDIF, were added to the standard syntax of the ON-GOTO and IF-THEN-ELSE statements. The translator implements the three additional statements by replacing MCI's blocked structures with combinations of standard Microsoft IF-THEN and GOTO statements. This produces an intermediate object program that is standard Microsoft BASIC. The two examples in Table I (at right) show how the IF-THEN-ELSE-ENDIF and REPEAT-UNTIL statements are translated.

Line numbers are replaced with labels in the TRANSFORM source language. A label is any combination of alphanumeric characters up to 255 characters in length that is preceded by the character "@". The translator generates the line numbers that are required by Microsoft BASIC when the intermediate object file is created. A symbol table can be produced

that contains the label to line number mapping. A restriction: only comments may follow a label definition, since the translator turns all label definitions into REMarks for documentation in the intermediate object program.

MCI added an INCLUDE facility to allow any file that contains BASIC statements to be copied into a program. A DEFINE statement may be used in conjunction with an INCLUDE statement to redefine the variable names in the included source. This allows a programmer to use the same procedure in many programs, or several times in a single program, without modifying the variables used in the procedure. Since all BASIC variables are global, using a facility such as MCI's INCLUDE is as close as you can get to having local procedures in BASIC. The INCLUDE statement may be nested up to five levels deep.

TRANSFORM allows a programmer to enable and disable the translator from within the program. Two special labels, @PREON and @PREOFF, are used for this purpose. When the translator is disabled, the standard Microsoft BASIC ON-GOTO and IF-THEN-ELSE statements can be used. @PREON and @PREOFF

function correctly with the INCLUDE statement by only affecting the level of INCLUDE at which they were specified.

The TRANSFORM Translator

The TRANSFORM translator consists of two COM files, TF.COM and TF1.COM. To invoke the translator you need only enter TF. The program prompts the user for all information that is needed to perform the translation. Most options have a default response that can be chosen by entering a carriage return. Default file name extensions of PRE and BAS are assumed for the source input and object files. TRANSFORM searches for the source if it is not on the specified drive. The order of the search is the currently logged-in drive, followed by the system drive. TRANSFORM's processing options include line number and increment selection for the translated source, listing and error message routing, hardcopy formatting information, and symbol table creation.

Once file name and option selection is complete, TRANSFORM begins its first pass on the source file. This pass translates the structured BASIC source program to standard Microsoft BASIC. The

TRANSFORM Source	Translated Object Code
IF (condition) THEN (block of statements)	1000 IF NOT (condition) THEN 1200 (block of statements)
ELSE (block of statements)	1190 GOTO 1400 1200 REM (block of statements)
ENDIF	1400 REM
REPEAT (block of statements)	1000 REM (block of statements)
UNTIL (condition)	1200 IF NOT (condition) THEN 1000

Table 1

Examples of how the IF-THEN-ELSE-ENDIF and REPEAT-UNTIL are translated.

translator may be interrupted at any time. Once interrupted, you may continue, translate a different source file, or terminate the translator. At the completion of pass one, TRF.COM is loaded and pass two begins. This pass performs all the line numbering that is required by Microsoft BASIC. When the translator has finished, an intermediate object program has been created which can be loaded by MBASIC or compiled by BASCOM.

TRANSFORM programs must be created with a text editor, since BASIC editors always require line numbers. Some text editors, such as Wordstar, set the high order bit on certain characters. This causes problems for the translator, so a utility named SIFT is provided to strip off all the high order bits in a text file. This utility is not needed if your text editor doesn't set the high order bit on any ASCII characters.

Documentation

The documentation that is provided by MCI is divided into two sections, a tutorial and a user's manual. The tutorial demonstrates how the translator is invoked and describes the procedures that are needed to get from MCI-enhanced BASIC to Microsoft BASIC. An additional feature in the tutorial is a discussion of structured programming and how it can be used to simplify the task of creating an "error-free" program. As an example, the tutorial proceeds step by step through the design of a program to format and print BASIC program source files. The source code for the example program is included, so that the user may experiment with TRANSFORM without first having to write a program. Besides being the basis for the tutorial, the print program is a useful utility.

The user's manual is a logical progression of information about TRANSFORM. The first part of the manual contains general information on the purpose of TRANSFORM and its theory of operation. The second part details the syntax of all aspects of MCI's enhanced BASIC language.

Reviewer's Comments

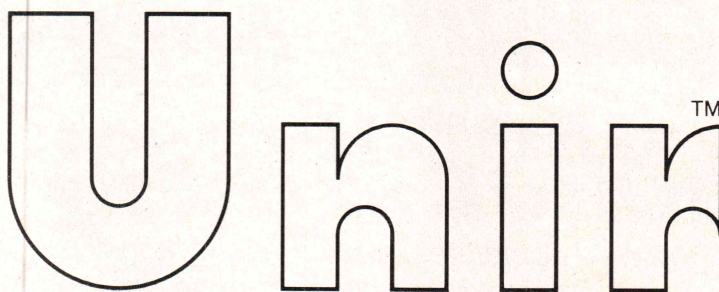
TRANSFORM allows a programmer to write BASIC programs using blocked control structures that have been present in other high-level languages for years. Ideally, there would be no use for such a product, as people who wished to write structured programs would use an available high-level language. Unfortunately, due to the number of years Microsoft BASIC has been around, the availability of a compatible compiler, and the costs associated with purchasing several language processors, this product does serve

a useful purpose. Many CP/M software products also require Microsoft BASIC, such as MCI's MCDISPLAY, making the use of a structured language impossible.

One disadvantage to MCI's approach to structured BASIC is that it adds another step in the process of creating an executable program. It is a costly step regarding compile time, as translating a 500-line program takes between eight and ten minutes. There is also potential for problems in the translated source due to the way TRANSFORM implements its blocked structures. In Microsoft BASIC any non-zero expression is considered to represent the true condition for an IF statement. MCI uses "IF

NOT(condition) THEN" as part of the translation for each of the blocked structures. This creates the restriction that all conditionals used by the blocked control structures must evaluate to true (minus one) or false (zero). Although this restriction is well-documented, the burden is on the user to make sure that it is followed.

MCI's product TRANSFORM does work. The documentation is complete and the translator is easy to use. If you already have Microsoft BASIC, it would be difficult to find a structured language interpreter or compiler of this caliber for \$39.95.



The Unir Project is a high technology user's group for people interested in the UNIX operating system and the C programming language. Members of The Unir Project have access to a wealth of information through The Unir Project Technical Reports and Newsletters. Most of the reports and newsletters contain complete **SOURCE CODE LISTINGS** written in C. Many of the programs can be adapted to non-UNIX environments that support the C programming language.

Membership in The Unir Project is open to anyone interested in interactive computer systems. Each member is assigned a Universal Numeric Identifier (UNID) that is certified to be unique. The UNID is used when ordering reports and software from The Unir Project. The UNID is also used when accessing the data base of members.

Each new member receives a certified UNID, a membership certificate, an id card, a copy of the current newsletter, a souvenir UNIKEY, and a listing in the online member directory. Additional newsletters, reports and software can be ordered once you are a member.

To become a member send your name, address, and phone number plus a brief description of your interests to:

Unir Corporation
5987 E. 71st St., Suite 106
Indianapolis, IN 46220

Voice: 317-842-7014
Data: 317-842-6986 (7 Bit NAPLPS)

Enclose \$25.00 in check or money order to cover your initial membership. Indiana residents add 5% sales tax. Please allow 4 weeks for delivery.

UNIX is a trademark of Bell Laboratories
Unir, UNID, and UNIKEY are trademarks of Unir Corporation

Circle no. 116 on reader service card.

COMM-X-PAC

Company: Hawkeye Grafix, 23914 Mobile, Canoga Park, CA 91307

Computer System: CP/M 2.0 or greater

Price: COMM-X-PAC: \$150, or COMM-X only: \$99

Circle Reader Service No. 123

Reviewed by Gene Head

I write for several computer magazines and have a listed phone number. So I suppose I get more than my normal share of phone calls from individuals with questions about microcomputers. Eight out of ten callers are requesting help getting a modem working. Seven of those

eight are non-programming types trying to get some version of Ward Christensen's MODEM program working. For many, it is a simple matter to set a program's equates to a serial RS-232 status and data port and to monitor the proper status bits. For just as many this task is impossible.

Will Pierce at Hawkeye Grafix has taken on the task of delivering a modem communications and control package that is factory patched to run without change on the user's computer system. When ordering COMM-X-PAC, the user specifies his/her computer and modem, and Hawkeye does the rest. The purchaser receives a disk with programs that run without the need for any adaptations. Well almost...

The package I ordered for my IMSAI MIO serial I/O card was delivered with the status bit logic reversed. In other words, when executing a status check, the instruction JZ (jump if zero) should have been JNZ (jump if not zero). I know the MIO card is working properly and wired per the manufacturer's specifications because the firmware monitor in a factory ROM controls the MIO card without a hitch. Now it should be noted that IMSAI has been out of business for several years and expecting a vendor to support equipment which is no longer supported by even the manufacturer could be asking too much.

The package my company ordered for a brand-new Molecular Super Micro 30 had two fatal bugs. Molecular's N/Star operating system uses an expanded and larger-than-normal CCP. Fortunately, this package was purchased as assembly language source code. I had no trouble tracking down three places in the source code that expected the normal sized CCP and making the proper modifications. The second error surfaced when the "phone is ringing" response from the smart modem caused the activity time-out timer to reset. This was fixed by telling the smart modem not to send responses after a caller connected.

COMM-X-PAC includes, in addition to COMM-X and CONSOLX (discussed below), a "data base management" program and a message board program both written in Hawkeye Grafix compiled VBASIC. Neither of these programs performs as well as the public domain programs available on larger RCPM systems and through user's groups.

COMM-X-PAC is two separate packages: COMM-X for connecting to a host computer like the Source or an RCP/M and CONSOLX for making your computer act like a host or RCP/M.

COMM-X

COMM-X is the terminal program that allows your computer to communicate with other (host) computers. COMM-X is far superior to even the most advanced public domain MODEM program that I have seen to date. It can virtually communicate with any system like IBM TSO, HP-3000, as well as RCP/M's. It has keyboard macro definitions; an entire call, connect, and log-on procedure can be compressed to one or two keystrokes. A half-dozen separate programs allow for an automatic dial-up directory, encoding and decoding private files and un-loading binary files to INTEL HEX format files for systems that are restricted to sending and receiving only seven bits of data.

Hawkeye has included Dick Greenlaw's squeeze and un-squeeze programs among its utilities. These can cut modem connect time by 50% by compacting the

"Q-PRO 4 blows dBASE II away

We now complete complex applications in weeks instead of months."

says Q-PRO 4 user, Richard Pedrelli, President, Quantum Systems, Atlanta, GA

"As a dBASE II beta test site the past two years, we were reluctant to even try Q-PRO 4. Now we write all our commercial applications in Q-PRO 4. We find it to be an order of magnitude more powerful than dBASE II.

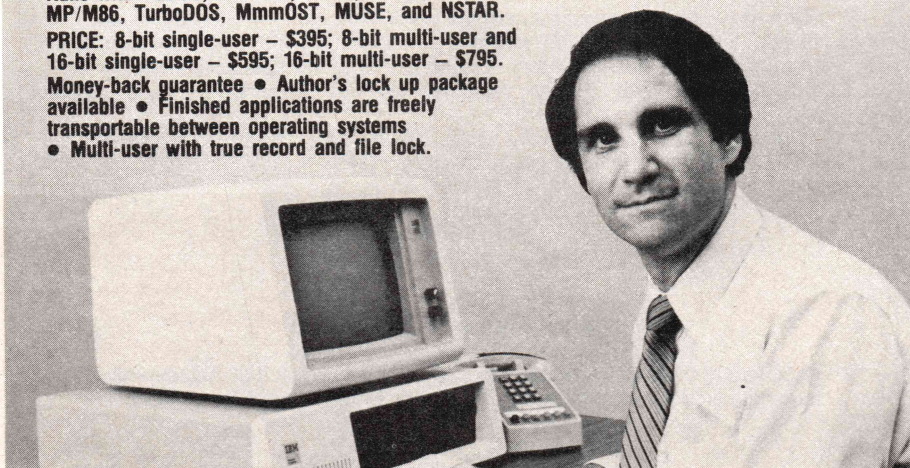
We used Q-PRO 4's super efficient syntax to complete our Dental Management and Chiropractic Management Systems much faster. Superb error trap and help screen capabilities make our finished software products far more user friendly, too.

In my estimation, any application programmer still using outdated 3rd generation data base managers or worse, a 2nd generation language like BASIC, is ripping himself off."

Runs with PC-DOS, MS-DOS, CP/M, MP/M, CP/M86, MP/M86, TurboDOS, MmmOST, MUSE, and NSTAR.

PRICE: 8-bit single-user - \$395; 8-bit multi-user and 16-bit single-user - \$595; 16-bit multi-user - \$795.

Money-back guarantee • Author's lock up package available • Finished applications are freely transportable between operating systems • Multi-user with true record and file lock.



For Q-PRO 4 demonstration, go to nearest MicroAge store or other fine dealer.

quic-n-easi products inc.

136 Granite Hill Court, Langhorne, PA 19047 (215) 968-5966 Telex 291-765

CP/M, MP/M, CP/M86, and MP/M86 are trademarks of Digital Research. TurboDOS, MmmOST, MUSE, NSTAR, MS-DOS and PC-DOS are trademarks of Software 2000. TeleVideo Systems, D.S.M., Molecular, Microsoft and IBM, respectively.

file to be sent, sending the compacted version and then un-compacting it at the receiving end. There is also a sorted disk directory utility but this utility is not as impressive as the public domain program SD-44.COM (Super Directory) described in The CP/M Exchange column of *DDJ*, June 1983.

COMMx supports two error-checking protocols and two methods of file transfer. The COMMx-to-COMMX file transfer system seems to be more advanced than the COMMx-to-MODEM7 system. Earlier versions of COMMx didn't support the Christensen protocol and I'm glad to see the newer version does.

The most advanced feature of COMMx is the E-MAIL feature. Using the dial-up directory, keyboard macros, and a submit file, COMMx can establish communication with other COMMx/CONSOLX sites and exchange mail files while each site is unattended. The entire process can be initiated and then delayed a specified number of hours and minutes to take advantage of off-hour phone charges. COMMx continues trying to establish communications until the job is completed. This E-MAIL requires the most effort to get operational, but the manual is clear and well laid out in describing the E-MAIL features.

The manual also includes an index and even a glossary for non-computing types to refer to. Eighty percent of the manual is dedicated to COMMx with the remaining twenty percent devoted to CONSOLX.

CONSOLX

CONSOLX, like COMMx, is configured at the factory to the user's machine and modem. The list of supported machines and modems is long and growing. I guess nine of ten personal as well as business micros are currently supported.

CONSOLX allows your computer to function as a host to callers accessing your system. Field selectable options include a password, ring-back operation, a welcome message, and UART configuration to either 7-bit even parity or 8-bit no parity. An auto start feature allows your system to execute a program after the user gets past the optional password and welcome message.

At the Money Tree (the company I work for) we run a system that logs callers in, gives them access to proprietary financial information, and logs them off, keeping track of time and charges. Because of its ability to answer the phone and then autoloading and execute other programs, CONSOLX works well as the base system. Callers see only as much of the system as they need to see. I know of at least one other business using CONSOLX commercially.

Conclusion

I had to write the routines to switch baud rates in the CONSOLX program and I think these should be included in the package. Also, working without carrier detect requires disorderly log-offs to time out for five minutes before the next caller can connect. I was able to get around this on the Molecular, but I'm not sure this problem can be solved generally for all systems.


For some of us, getting a program to work is half the fun of computing. For others who want it to work the first time and every time, COMMx-PAC from Hawkeye Grafix is clearly worth the \$150 price tag. Furthermore, if you have no need for making your computer a host

computer and only want to originate calls to other host systems, the COMMx program alone sells for \$99.

I commend Hawkeye Grafix for an outstanding package in COMMx-PAC. They have taken on a tough job of providing software for various hardware environments, and as far as I know they are doing the job well. My calls to Will Pierce have always been given professional yet personal treatment. I can personally recommend all their fine products, especially the COMMx-PAC communications package.

DDJ

GOOD NEWS!



C for the 6809 WAS NEVER BETTER!

INTROL-C/6809, Version 1.5

Introl's highly acclaimed 6809 C compilers and cross-compilers are now more powerful than ever!


We've incorporated a totally new 6809 Relocating Assembler, Linker and Loader. Initializer support has been added, leaving only bitfield-type structure members and doubles lacking from a 100% full K&R implementation. The Runtime Library has been expanded and the Library Manager is even more versatile and convenient to use. Best of all, compiled code is just as compact and fast-executing as ever - and even a bit more so! A compatible macro assembler, as well as source for the full Runtime Library, are available as extra-cost options.

Resident Introl-C/6809 compilers running under Flex or OS9 are priced from \$375; Uniflex, from \$425.

Cross-compilers for PDP-11/Unix hosts are priced from \$1500.

Trademarks:
Introl-C, Introl Corporation;
Flex and Uniflex, Technical Systems Consultants;
OS9, Microware Systems;
PDP-11, Digital Equipment Corp.;
Unix, Bell Labs.

For further information, please call or write.



647 W. Virginia St.
Milwaukee, WI 53204
(414) 276-2937

Circle no. 52 on reader service card.

An Introduction to Programming and Problem Solving with Pascal, Second Edition

By G. Michael Schneider, Steven W.

Weingart, and David Perlman

Published by John Wiley & Sons, Inc.

\$21.95, 468 pages

Reviewed by William G. Hutchison, Jr.

"What? Another Pascal textbook?"

If your reaction was the same as this reviewer's, please do not give up in despair. This book is not another quick-and-dirty attempt to cash in on the demand for computer texts; it is a thoroughly professional effort. It comes from that hotbed of Pascal activity, the University of Minnesota. Andy Mickel read the proof copy. You may remember him as the former editor of *Pascal News*. Niklaus Wirth was interested enough in the project to write a foreword.

This is clearly a college text. The authors wrote it to be used as text for the course CS I, Computer Programming I, of ACM Curriculum '78. They have achieved a very good balance of subject material. Some authors overload their readers with thorny calculus problems to be programmed. This is useful material, but not suitable for an introductory programming course. This text is evenly balanced between text processing applications, commercial (payroll and inventory), games (LIFE), and elementary statistics. The authors use real programs as examples, not meaningless fragments cooked up just to display some feature of the language.

The authors follow their own advice, which is to plan what to do before beginning the coding. Chapter 1 is devoted to problem specification. They point out correctly that most program errors start from errors or ambiguities in the statement of the problem, so it is worthwhile to spend considerable time getting it right at the beginning. Chapter 2 is about developing algorithms in pseudo-code. It contains a brief discussion of program efficiency and asymptotic analysis (e.g., big-O function, n -squared versus $n \log n$ performance), not enough to bog down the reader, but enough to stimulate the student to take the next course.

In addition to the usual programming material, this book includes a chapter on modular program design, and one on program debugging, testing, documentation, and maintenance. Unlike some authors, they do not succumb to the illusion that structured programming means no errors

in programs! That is what makes this book more than just another textbook. The technical material is well organized and presented, and it also contains a great deal of practical wisdom, which will save the careful reader from many unnecessary mistakes.

The Pascal language used here is that of the Jensen & Wirth text, *PASCAL Users Manual and Report*, Springer-Verlag, 1974. It is presented in syntax graphs inline in the text as needed, and complete in an appendix. The timing of the text may be unfortunate. It is a rewrite of a 1978 edition. At that time, there was no official standard for Pascal, so it contains no reference to ISO standards or to conformant array parameters. That is not a serious problem for an introductory text, because the core of the language is almost unchanged since 1978. There is practically no reference to UCSD Pascal or other non-standard versions. There is a brief discussion of versions with INTERACTIVE files (which do not do a GET first in READ), and a note about versions with direct access files.

The only thing missing from this excellent text is a discussion of quicksort, which other authors have managed to present in an elementary manner. Let's not produce another generation of programmers who know only bubble sort!

Language Translations

By John Zarrella

Published by Microcomputer

Applications, Inc.

\$14.95, 200 pages

Reviewed by Michael Carter

Not the least of the advantages of the microcomputer revolution is the encouragement and indeed the incentive it gives the owner of a personal computer to delve into its workings and operations. This will inevitably lead to the spread of computer know-how from a select few to a wider audience — a development which I for one heartily applaud. It also implies the existence of a market for introductory computer texts, of which this book is a good example. The author of such a book faces a considerable problem — how to aim at an easy presentation without oversimplification. The current author is partially — but only partially — successful.

No user of a personal computer can avoid forming an intimate acquaintance with at least one language translator (most

commonly BASIC). A language translator is a computer program which accepts a human-oriented source language as input, translating it into a form which can be directly executed by the computer. It is the medium through which the user conveys his or her wishes to the machine.

After a general introduction, Zarrella deals in turn with the three main types of language translator: assemblers, interpreters (e.g., APL, BASIC, Forth), and compilers (e.g., C, COBOL, Pascal, Fortran). He appropriately emphasizes two important points. First, interpreters and compilers are not distinct categories but rather the ends of a continuum with most language translators lying somewhere in-between. The reader will thus understand why a BASIC program is stored in the machine in a form which he or she cannot readily read. The second important point is that the basic tasks required of a language translator are the same whether it is an assembler, compiler, or interpreter. Consequently, assemblers, compilers, and interpreters are made up of essentially the same functional components.

Zarrella makes convenient use of the latter point in the second half of the book when he deals in successive chapters with the functional components of any language translator: lexical analysis, syntax analysis, semantic analysis, code generation, and optimization. The lexical analyser examines the source text and divides it into its component parts: symbols, numbers, punctuation, and so on. The syntax analyser parses the input program to verify that it is a valid program in the language and to provide an appropriate framework for the next step. This is semantic analysis which determines the meaning of the source program. Code generation produces the actual executable machine code. Finally, optimization makes minor changes in the executable machine code to increase execution speed and/or decrease storage requirements. It should not be concluded that these five components of a language translator can necessarily be distinguished as separate entities that follow one another in the order given. They are conceptual divisions, and with the exception of optimization, they must be provided for in every language translator. The five constituent parts provide an appropriate framework for approaching the general task of language translation.

The five chapters dealing with the constituent parts of a language translator

are preceded by a chapter on "Languages and Grammars." This is the theoretical core on which language translation, in particular syntax analysis, is based. Unfortunately, this is not easy material to grasp and the chapter will probably require more than one reading. It is here and in the chapter dealing with syntax analysis that Zarrella's exposition falters. I think this is because he is trying to compress some complex ideas into two short chapters. Unfortunately, this material is really the core of the matter and it left me feeling unsatisfied. I think some more material would have been justified. Further, I found his use of the analogy of a divisionalized company to explain top-down parsing overdone and superfluous.

There are three supplementary chapters which have not yet been mentioned. One deals with program development utilities: linkers, loaders, library managers, etc. A second deals with macros, which are most commonly (but not exclusively) found in conjunction with assemblers. A third deals with the important topic of error processing. A glossary, a short bibliography, and an index round out the book.

This book has an admirable objective: to present the general concepts of language translation, emphasizing the similarities between the different classes of assembler, interpreter, and compiler in a manner accessible to the neophyte. This is a very ambitious task and one in which the author is not entirely successful. I suspect that it is more likely to whet the appetite than satisfy it. Hopefully, it will encourage the reader to seek further information in one of the more substantial texts. To any computer user who is curious about what lies behind the cosy facade of a BASIC interpreter or wonders why Pascal can't be like BASIC, I suggest this book as a good place to start.

Z80 Assembly Language Subroutines

By Lance A. Leventhal and
Winthrop Saville

Published by Osborne/McGraw-Hill

\$15.95, 498 pages, paperback

Reviewed by Paul J. Gans

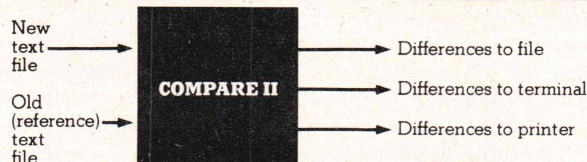
Learning assembly language for a new computer is not easy for anyone. Even experienced programmers go through a period of adjustment when faced with a new machine. Novices are urged to study the architecture of the new machine to learn its op codes and addressing modes and to struggle as best they can to write simple routines that have been written countless times before.

More experienced programmers know that there is more to it than that. They do not become comfortable in the machine language of a new machine until

No more lost edit changes!

COMPARE II

High performance differential text analyzer!



Writers! Researchers! Lawyers! Engineers! Programmers!
Use highly rated **COMPARE II**. Cut text analysis from hours to minutes!

- ▶ PC-DOS, CP/M-86 or CP/M 2.2
- ▶ Scans by word or by line
- ▶ Fast New Algorithm, No file restrictions
- ▶ You can customize for word processor, printer width, file defaults, specific work flow, computer languages, different highlighting techniques
- ▶ Clear commands, Numerous formatting options
- ▶ Can generate new document with change bars

Specify When Ordering: Operating System, Computer Type and Disk Format. Free brochure and *nearly* free demo disk available.

COMPARE II

Demo Disk (credits to purchase)..... \$12.95

\$145.00

**SOLUTION
TECHNOLOGY, INC.**

"We Deliver Productivity"

1499 Palmetto Park Road
Suite 218
Boca Raton, FL 33432
305/368-6228

PC-DOS and CP/M are trademarks of IBM and Digital Research respectively.

Check or COD, Florida residents add 5% sales tax. Dealer and Distributor inquiries welcome.

Circle no. 98 on reader service card.

Introducing SPL the first multi-mode spooler for CP/M computers

If you believed that your computer couldn't do better than a single task system think again. You can convert your machine into a dual-task computer with **SPL**, the amazing Spooler program developed by Blat R+D. **SPL** enables you to use hidden capacity available on your CP/M computer to print documents and run your ordinary programs, all at once.

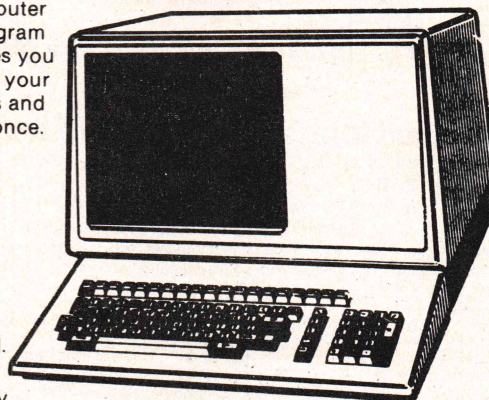
While printing, your regular programs won't stop processing, waiting for the printer to finish. **SPL** will store the information to be printed in internal or external (disk drives) memory until the printer is ready to receive the data. Result: your programs will run at full speed.

As **SPL** can use up to the full capacity of your disks for temporary storage, it's much more powerful than hardware spoolers, which are limited to 64k memory or less.

SPL is an advanced product with several modes of operation. In addition to intercepting the output to the printer, **SPL** can print your existing text files, or those that your programs will create from now on. **SPL** will even take care of tab expansion. As an added bonus, **SPL** needs no installation on most CP/M 2.x computers.

If you have a computer

Get A Second Computer FOR \$139



You could get an equivalent increase in computing power by spending \$1000 to \$3000, but **SPL** is only \$139, including disk and manual.

To order your **SPL** program call us today specifying what disk format you require. You can charge it to your VISA or Master Card if you prefer.

Blat Research+Development Corp.
8016 188th SW, Edmonds WA 98020
Call toll-free 1-800-LOBO-BAY
In WA call [206] 771-1408

Registered Trademark of Digital Research

they have mastered its idioms. *Z80 Assembly Language Subroutines* is a book of idioms for the Z80. It is a wonderful book that can be read with profit by anyone not yet able to speak idiomatic Z80.

The book begins with an explanation of Z80 architecture (with a section for experienced programmers, pointing out distinguishing features and difficulties with the Z80) and the Z80 instruction set. The instruction set is presented from a programmer's point of view. Thus instructions are grouped according to function. For instance, under "Storing Registers in Memory" follow direct storage of registers, indirect storage of registers, indexed

storage of registers, etc. The instruction set is covered fully with many examples.

Chapter 2, entitled "Implementing Additional Instructions and Addressing Modes," is filled with instances of common operations requiring more than a single instruction. These operations occur repeatedly in real programming. The novice must, each time, figure out how best to perform them. These operations are, in fact, idioms, and are at the fingertips of experienced programmers. For example, how does one compare the contents of the HL register pair to a 16-bit immediate value? On page 81, Leventhal and Saville give two methods, one requiring

two instructions, the other, three. Chapter 3 is devoted to common programming errors and includes such topics as using the flags incorrectly, confusing registers and register pairs, and handling arrays incorrectly.

The bulk of the book, however, is devoted to assembly language subroutines. There are more than 50 of them, covering topics such as code conversion (binary to BCD, ASCII decimal to binary, EBCDIC to ASCII, etc.), array manipulation, arithmetic (multiple-precision binary division, etc.), string manipulation (compare, concatenate, etc.), array operations, input/output, and interrupts. Each subroutine listing is elaborately commented, entry and exit conditions are clearly given, and execution times (in machine cycles) are provided. There is an introductory explanation of the function of each subroutine and a description of how it works. Best of all, test data and results are given. Much can be learned by examination of these subroutines, and, better yet, they are useful.

The Z80 instruction mnemonics used are Zilog's, which I believe are superior to the multiple and incompatible sets of pseudo-8080 mnemonics in common use. All notational conventions are explained. Appendices contain a detailed Z80 instruction set summary with reproductions of the Zilog instruction charts, a programming reference to the Z80 PIO, and an ASCII character set chart. The book concludes with a 23-page glossary and a useful index.

Mathematical Theories of Optimization; Proceedings, S. Margherita Ligure 1981; Lecture Notes in Mathematics Series No. 979
 Edited by A. Dold and B. Eckmann
 Published by Springer-Verlag
 \$14.50, 268 pages, paperback
 Reviewed by Robert Irving

Optimization is the science of obtaining the maximum return for investment of resources under a given set of conditions. The field is also known as "operations research" from its initial application in World War II. The most common applications lie in the areas of inventory control, product selection, and manufacturing scheduling.

Engineers and computer scientists in the field of optimization are usually looking for applied solutions to real problems. They generally want either an algorithm or a program which will solve the problem at issue. Mathematicians, on the other hand, are not necessarily looking for solutions as such. They are more interested in the nature of the statement of the problem. Such things as "existence theorems"

get it
in gear
with

Cache/Q™

THE Software Accelerator

by techne'

GET YOURS MOVING NOW!

Why wait longer, to buy it ... to try it?

- speeds up your programs, saves time and money
- faster dBase II, Wordstar and many more
- makes your computer investment really pay
- gets your computer in gear ... gets you in gear

Cache/Q features automatic memory management for your IBM PC providing:

Disk-buffering that can speed up your application programs up to four or more times the speed you are now getting.
 Background print-buffering that will allow you unrestricted use of your computer with concurrent operation of up to five attached printers, each with an independent data stream.
 Automatic write-thru to your disks protecting your working data from power failure.
 Capability for specifying selective buffering of data according to disk-drive, file, even extension.

VERSATILE • FLEXIBLE • EFFICIENT • EFFECTIVE

To order CALL NOW: (800) 835-9595. In Calif. (800) 835-9594—Dealer Inquiries Welcome

IN CANADA CONTACT: CYBERTECH TRADING CORP.
 Unit 9, 11720 Voyageur Way • Richmond, B.C. V6X 3E1
 Telephone (604) 278-4848 • Toll Free 1(800) 663-9767

Techne' Software Corp. • 3685 Mt. Diablo Blvd., Suite 210 • Lafayette, CA 94549

Cache/Q is a trademark of Techne Software Corp. • Cache/Q is a product of Techne Software Corp. • Copyright© 1982 by Techne Software Corp.
 CP/M is a registered trademark of Digital Research, IBM and PC-DOS of IBM Corp., dBase II of Ashton-Tate, Inc., Wordstar of MicroPro, Inc.

Cache/Q Specifications

OPERATING SYSTEMS:
PC-DOS

(Also Available for CP/M 2.2)

BUFFERING TECHNIQUE:

Least Recently Used (LRU) algorithm controls disk-buffering; automatic disk/print buffering dynamically adjusted for highest efficiency.

BUFFERING CAPABILITY:

Utilizing up to 590K of memory, Cache/Q will operate with up to 16 attached mass storage devices, with sector sizes ranging from 128 bytes to 16K bytes and can operate up to 5 attached printers.

INSTALLATION:

No modification of existing programs, operating procedures, or hardware required. A simple interactive configuration program needs no special expertise to operate.

(does a solution exist?) and "robustness" (does the solution apply over a broad range of conditions?) are of more import than a particular solution.

This book is pure mathematics with potential practical application. Consequently, it is of little direct interest to most computer programmers.

Ultimately, some of the articles may lead to algorithms of great importance. For example, the article by L. Cesari on "Existence of Solutions and Existence of Optimal Solutions" may someday encourage other mathematicians to produce algorithms with which we can optimize some problems not now covered, but possible.

On the other hand, I see little to no potential use for a discussion by A. Bensoussan on page 71, entitled "On the Production Smoothing Problem," in which he extends a discrete case to a continuous case. First, I know of no "continuous" production process. Even oil refineries, which work with large volumes of material, operate on a "batch" basis. No two batches have the same characteristics, hence the run will be "continuous" only for the length of the batch. Second, all of us who make mathematical models on the computer know that we are dealing with discrete processes, since that is the only kind of process the computer can deal with.

Unless you are a "pure" mathematician interested in the rigors of that field, I suggest you forgo buying this volume.

Teletext and Videotex in the United States

By John Tydeman, Hubert Lipinski,
Richard P. Adler, Michael Nyhan,
and Laurence Zwimpfer

Published by McGraw-Hill
\$30.00, 314 pages

Reviewed by Susan Bowers

Teletext (one-way information services) and Videotex (two-way information services) are new technologies created by the marriage of computing and communications. They emerged in England and France in the 1970's, have been growing rapidly in Europe, Canada, and Japan, and are currently becoming important tools in this country.

Teletext and Videotex in the United States is a look at Teletex and Videotex from a futurist viewpoint. Using careful research and documentation, this in-depth overview deals with three classes of services: public data bases, closed user groups, and private systems. Full of charts and tables, the book looks at the present and future states of this technology in the United States. The first five chapters deal with defining the phenomenon, detailing

its history, assessing the current state of information services, and discussing possible future applications. The rest of the book develops a model which studies the possible extent and impact of Teletex and Videotex in the United States by the end of this century.

The book emphasizes such public policy concerns as the effect of those technologies on society, legal aspects, security, privacy, standards, consumer protection, international ramifications, advertising liability, and impartiality. The modeling and forecasting are organized and carefully done, covering the possible futures

from several perspectives. There is an impressive appendix on policy issues and an extensive bibliography, as well as a comprehensive index.

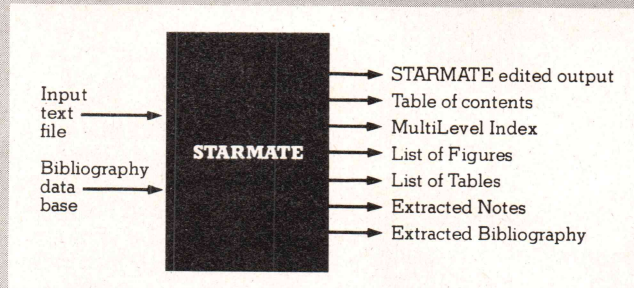
Teletext and Videotex in the United States provides an interesting look at future trends in the information services. Several parts of it would add much to a study of computers in society; it could also be an excellent textbook for a course on the future impact and direction of information services in the United States.

DDJ

Get more than WordIndex for less \$!

STARMATE

Users get high speed document finishing for WordStar with STARMATE under PC-DOS, CP/M-86, or CP/M 2.2.



Writers! Researchers! Engineers! Cut document makeup time from hours to minutes!

- ▶ Multi-Level Table of Contents
- ▶ Multi-Level Index
- ▶ Lists of Tables and Figures
- ▶ Numbers Paragraphs (1.2, 1.3, etc)
- ▶ Prepares Footnotes
- ▶ Prepares Bibliography
- ▶ Clear commands, Numerous formatting options
- ▶ Reads documents with nested files

Specify When Ordering; Operating System, Computer Type and Disk Format. Free brochure and nearly free demo disk available.

STARMATE

(Special Introductory Price)

Demo All Disk (credits to purchase) \$12.95

\$145.00

SOLUTION TECHNOLOGY, INC.

"We Deliver Productivity"

1499 Palmetto Park Road
Suite 218
Boca Raton, FL 33432
305/368-6228

WordStar, PC-DOS and CP/M are trademarks of MicroPro, IBM and Digital Research respectively.

Check or COD, Florida residents add 5% sales tax. Dealer and Distributor inquiries welcome.

Circle no. 97 on reader service card.

by Michael Wiesenberg

Stop Noise

Kalglo's IPC international model computer power control console with surge suppression and power line noise filter has six individually switched IEC-320/CEE-22 outlets in dual separately filtered banks of three each. It absorbs 318 joules at 18,200 amp pulse, is rated at 10 amps maximum with 100 or 260 VAC operation, 50 or 60 Hz, weighs 2.5 pounds, and costs \$150. **Reader Service No. 101.**

The Lemon AC Surge Protector from Electronic Protection Devices gives protection in excess of 6000 volts and 200 amps, with instantaneous voltage clamping including grounding to prevent flashover to other ports and damage to equipment. The company thinks its product is so good that insurance by Lloyd's of London is included, protecting user hardware up to \$2500 from damage caused by transient voltage surges. The Lemon lists for \$59.95. **Reader Service No. 103.**

Buffer It

Big-Buffer from Mikrocomputer-technik of Germany is a Centronics-compatible hardware spooler for parallel printers. The 8K version is \$170, the 32K version is \$195, the 64K is \$256, the 96K is \$317, and the 120K is \$363. You get all cables and connectors (36-pin Centronics with female input and male output). Power comes from the printer using pin 18 of the Centronics connector, or you can purchase a separate 110V, 60Hz power supply for \$30. These prices are based on the exchange rate at the time of the release, 2.60 DM to \$1 US, and are subject to change. **Reader Service No. 105.**

Can't Rip It

Micro Format has Flexi Disk Envelopes made of Dupont's Tyvek, a supposedly tear- and wear-proof substance that looks and feels like paper. They included a sample with their press release, and I couldn't tear the stuff. Each package has 25 envelopes, and you get five packages postpaid for \$25. **Reader Service No. 107.**

TRS-80 Assembly Language Programming

TRS-80/Z80 Assembly Language Library by Craig Lindley from WG-Books has 75 program listings to thoroughly explain assembly language on the TRS-80, and includes two TRSDOS disks with 33 source and object files, for \$34.97. **Reader Service No. 109.**

Dr. Dobb, Meet Dr. Logo

Dr. Logo, from Digital Research, is an advanced version of Logo for 16-bit micros, in particular the IBM PC and XT. It uses turtle graphics, just like the popular version of the language. Written in C, Dr. Logo is transferable to other systems. You get a larger work space, with over 10,000 nodes of memory, built-in help commands, a full-screen procedure editor (with on-line help as well). Also supported are comments, indentation, user-defined words, windowing, upper and lower case, new primitives, string processing, line editing, and debugging with a trace mode and a watch mode. The latter is for checking or modifying variables or expressions immediately after execution of a statement. Cross referencing, double-precision floating point, transcendental functions, and a \$149.95 price tag are part of the package as well. **Reader Service No. 111.**

Go Forth on PC

HSFORTH, from Harvard Softworks, is Forth-79 for the PC and other 8086 and 8088 machines. Word names, definition lists, and machine codes use separate segments, permitting programs up to 192K. HSForth runs the Sieve of Eratosthenes in 47 seconds, as opposed to 70 to 140 on other Forths, and 2000 for interpreted BASIC. Auto-Opt and the Micro-Asm extension reduce the time to five seconds. Even optimized machine code, according to Harvard Softworks, runs no faster than five seconds. \$220. **Reader Service No. 113.**

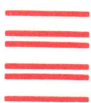
Things I've Tried

Usually the products I write about in this column are picked from a stack of press releases sent each month to *DDJ*. I rarely have the product in front of me when writing about it, and descriptions here are not intended to be endorsements.

Once in a while I do get the opportunity to try out products, and I'd like to share a few with you.

You may have gathered from previous columns that I like inexpensive computers, particularly the Timex products. One of the wonderful things you can do with a TS1000 or 1500 is learn Z80 assembly language. BASIC on a Timex runs achingly slow, particularly since some 60% of the microprocessor's time is spent updating the screen. Machine code programs, however, run literally hundreds of times as fast as BASIC ones. Several good books have been written about machine code programming on the TS1000. The best of these are two by Dr. Ian Logan, whose knowledge of the ZX80 and 81 (predecessors of the TS1000) is exceeded only by that of Clive Sinclair (inventor of these and the Spectrum, a version of which should now be available, modified by Timex and called the 2068).

You cannot write effective code for these little wonders without knowing the entry points for the various system variables. Melbourne House, an international publishing company, has Dr. Logan's *Understanding Your ZX81 ROM*, \$14.95, and *The Complete TS1000/Sinclair ZX81 ROM Disassembly*, \$19.95, and their own apparently authorless *Machine Language Programming Made Simple for Your Sinclair & Timex TS1000*, \$14.95. I have all three of these, and they are excellent. Melbourne also offers *Not Only 30 Programs for the Sinclair ZX81*, \$9.95, program listings that ingeniously all run in the "unexpanded" TS1000 (2K RAM), and what appears to be the ultimate hardware manual for the two "simpler" machines, *The Ins and Outs of the Timex TS1000 & ZX81*, by Don Thomasson, \$12.95. Add \$2 for p.&h., and residents of California, Maryland, and Tennessee add sales tax. Melbourne carries software and books for Timex, TRS80, Vic 20, and Commodore 64. They will send you a free catalog and



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

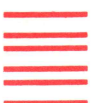
FIRST CLASS PERMIT NO. 756 MENLO PARK, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Dr. Dobb's Journal

For Users of Small Computer Systems

P.O. BOX E
MENLO PARK, CA 94026



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 756 MENLO PARK, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Dr. Dobb's Journal

For Users of Small Computer Systems

P.O. BOX E
MENLO PARK, CA 94026



Dr. Dobb's Journal

For Users of Small Computer Systems

Reader Service Card

To obtain information about products or services mentioned in this issue, circle the appropriate number listed below. Use bottom row to vote for best article in issue. This card is valid for 90 days from issue date. Use only one card per person.

December 1983, No. 86

Name _____ Address _____

Phone () _____ City/State _____ Zip _____

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81
82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108
109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162
163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189

Articles: 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214

Comments:

Purchase of Magazine:

- ☐ 1 Subscription
☐ 2 Computer Store
☐ 3 Newsstand
☐ 4 Bookstore
☐ 5 Passed on by friend/colleague
☐ 6 Other _____

Dr. Dobb's Journal

For Users of Small Computer Systems

Reader Service Card

To obtain information about products or services mentioned in this issue, circle the appropriate number listed below. Use bottom row to vote for best article in issue. This card is valid for 90 days from issue date. Use only one card per person.

December 1983, No. 86

Name _____ Address _____

Phone () _____ City/State _____ Zip _____

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81
82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108
109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162
163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189

Articles: 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214

Comments:

Purchase of Magazine:

- ☐ 1 Subscription
☐ 2 Computer Store
☐ 3 Newsstand
☐ 4 Bookstore
☐ 5 Passed on by friend/colleague
☐ 6 Other _____

PLACE
STAMP
HERE
The Post Office
will not deliver
mail without postage

Dr. Dobb's Journal

For Users of Small Computer Systems

ADVERTISING DEPARTMENT

P.O. BOX E

MENLO PARK, CA 94026

Dr. Dobb's Journal NOW AT BIGGER SAVINGS!

~~\$35.00~~
~~\$25.00~~

If you take advantage of this special offer you save over \$10 off newsstand prices, that's a 30% savings!

____ Please charge my: ____ Visa ____ MasterCard ____ American Express
____ Payment enclosed ____ Bill me later

Card # _____ Exp. date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____ Z6

Offer good in USA only. Foreign rates upon request. Please allow up to six weeks for first issue. This offer good until March 31, 1984.

A publication of People's Computer Co., P.O. Box E, Menlo Park, CA 94026.

PLACE
STAMP
HERE
The Post Office
will not deliver
mail without postage

Dr. Dobb's Journal

For Users of Small Computer Systems

ADVERTISING DEPARTMENT

P.O. BOX E

MENLO PARK, CA 94026

Dear Reader,

December 1983, No. 86

Dr. Dobb's has a long tradition of listening to its readers. We like to hear when something really helps or, for that matter, bothers you. In this hectic world of ours, however, it is often difficult to take the time to write a letter. This card provides you with a quick and easy way to correspond. Simply fill it out and drop it in the mail. We take care of the rest. Thanks for taking a few minutes to talk with us.

—Ed.

Which articles or departments did you enjoy the most this month? Why?
(Please indicate order of preference.)

Comments or suggestions: _____

Name _____

Address _____

"dBASE II® is far, far better than a shoehorn."

*Rusty Fraser
President
Data Base Research Corp.*

"We laughed when our customers asked us to put our minicomputer-based real-time accounting system, The Champion,TM on a micro.

"No way was it going to fit, we thought.

"We'd have to create our own database management system and, even then, it'd be a tight squeeze.

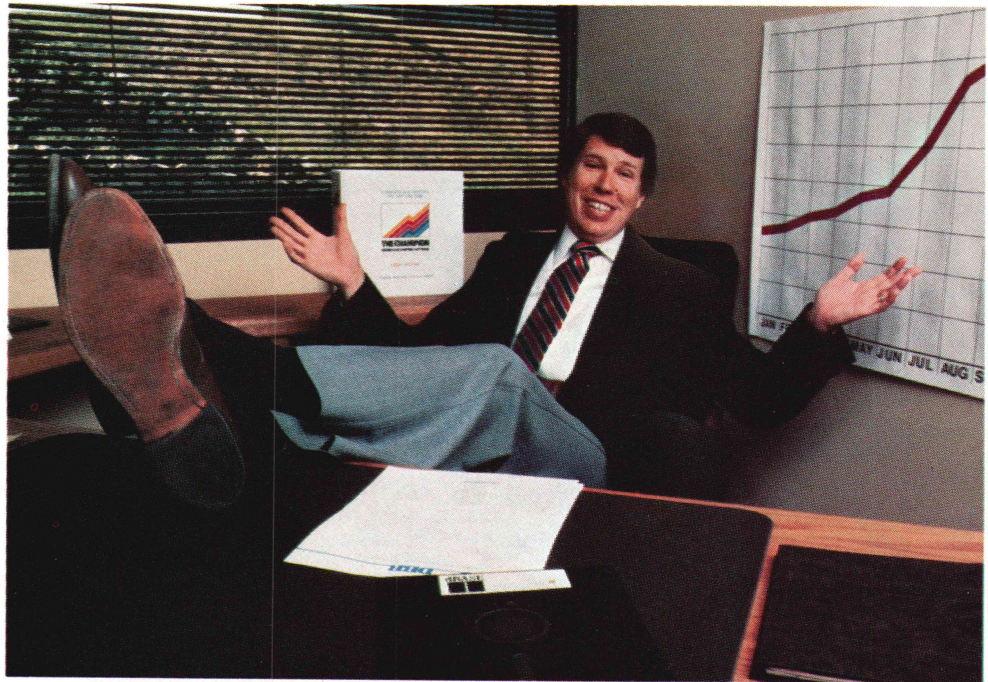
"Then we discovered dBASE II, the relational database management system for microcomputers from Ashton-Tate."

"dBASE II was a perfect fit."

"dBASE II is a program developer's dream come true. The dBASE II RunTimeTM module quickly provided us with the powerful text editing, data entry speed and other 'building block' capabilities we needed to develop and deliver a new Champion to our customers—the leading real-time on-line accounting system available for a micro."

The short cut to success.

The dBASE II RunTime module has helped a lot of program devel-



opers like Data Base Research become successful software publishers.

For more about dBASE II and RunTime, contact Ashton-Tate 10150 West Jefferson Boulevard, Culver City, CA 90230, (800) 437-4329, ext. 217. In the U.K., call (0908) 568866.

For more about The Champion, call Data Base Research at (303) 987-2588.

ASHTON · TATE

dBASE II and RunTime are registered trademarks of Ashton-Tate.
The Champion is a registered trademark of Data Base Research Corporation.
©Ashton-Tate 1983.

are looking for contributions from authors. **Reader Service No. 115.**

Far better than hand-assembling your machine code programs and then trying to POKE them into various memory locations is to use an assembler. International Publishing & Software Inc. offers **ZX Assembler**. It has its own editor. You write programs using standard Z80 mnemonics. No need to figure jump addresses; just use labels and the assembler figures them for you. Programs assemble in seconds and execute either directly from ZX Assembler's monitor mode or from a program that executes with a **USR** statement where the code is in a **REM** statement created by the assembler. The assembler also retains the source code in a second **REM** statement.

You can **SAVE** the entire program on tape, and alter, modify, and reassemble it. Those familiar with machine code on the TS will appreciate this method, because code loaded in other areas is either difficult to save, or cannot use relative addressing. In the monitor mode you can move whole blocks of code from one memory location to another, examine and modify any location of RAM, inspect and modify registers, and search the entire memory for any byte sequence. **ZX Assembler** costs \$14.95. International Publishing provides a free catalog and the name of the nearest dealer that handles this and other useful products, including **ZX Bug**, \$14.95, for debugging, editing, and running machine codes, **Toolkit**, \$14.95, to add nine powerful commands, including a renumber that

changes all **GOTO** and **GOSUB** references, **Fastload**, \$19.95, to load any program four to six times as fast, **ZX Forth**, \$29.95, for the ease of **BASIC** with machine code speed, and a stack of clever games. **Reader Service No. 117.**

If I knew that a product did not live up to its claims, I would not list it here. Usually I have no way of knowing that, because I just go by the press releases, which, naturally, are never unfavorable. Such was the case with Memotech, whose various memory modules and auxiliary keyboard for the TS1000 I cited a few months back. I believed their hype myself and shortly afterward bought their keyboard and 32K memory for my system. That first 32K memory didn't work at all, and I took it back to the store. The replacement 32K refused to work in conjunction with either Memotech's keyboard or the Timex printer. This was not much use to me since I needed the printer for listings during program development. I took my entire system to the store, and they could get the 32K memory and auxiliary keyboard to work with their systems only separately, not together.

I called Memotech several times for help, and they kept insisting that they had tested a setup identical to mine with no problems. Finally they agreed to look at my system if I would send it to them and, if either of their products was defective, to replace or fix it. I called them when the system was in their hands, and the technician to whom I spoke said he had set up my system and it had been running perfectly for ten minutes, and he could find nothing wrong with it. But while I was on the phone with him he said, "Wait a minute. It's failing." He conceded that there "might" be a problem with the memory module, which Memotech replaced. They then sent back my system.

When the system returned, again I could make the 32K memory work by itself only, not together with either the Timex printer or Memotech's keyboard. The keyboard, too, only worked by itself. The store from which I bought the memory unit was kind enough to take it back in exchange for a Timex 16K memory (and even refunded the difference in price!), with which I have had no problems. I sent the keyboard back to Memotech, requesting a refund, citing the "free ten-day trial period" clause in their warranty, stating that, while the ten days had elapsed all the problems had begun the instant I began using the keyboard and that I had been in contact with them right from the start. They refused to give me a refund, claiming that the refund applied only to those who had pur-

chased directly from them, but instead returned a replacement for my keyboard. If I wanted a refund, they told me, I would have to get it where I bought the keyboard.

The store, of course, did not want to give me a refund, because the 1500 would soon be released with its "real" keyboard, and auxiliary keyboards for Timex computers would soon be obsolete. They said that I should contact Memotech for a refund. (My guess is that Memotech is stuck with a warehouse full of these useless keyboards.) As I expected, the replacement worked no better than the original. At this point I no longer had their 32K memory, but the keyboard worked with neither Timex's printer nor the 16K memory.

By this time I had heard from the South Bay Area Timex/Sinclair Users' Group (mentioned in my October column) that Memotech was offering a free "modification" to purchasers of their 32K memories. It seems several other people had discovered that their memories did not work with Timex printers. Yet all the time I had been in communication with Memotech they insisted there was no incompatibility between their products and Timex printers; furthermore, they had never heard of anyone having any problems like mine. *Altogether, I went through three 32K memory modules and two keyboards without finding even one I could use.*

Now Dr. Dobb's has received a press pack announcing Memotech's \$595 **MTX-512 World Class Computer™**. It has a Z80A 4MHz processor with 80K standard RAM (expandable to 512K; 16K is dedicated to video). Among other things, it comes with Oxford BASIC and a language called **NODDY**, a real time clock, and a 79-key keyboard. I hope this product is better than the peripherals I couldn't use, and that others will encounter more competent service and better support and attitude than I did. I know I'll never get the chance to find out. **Reader Service No. 119.**

DDJ

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 207.

BASIC COMPILER NOW ONLY \$40 !

BASIC COMPILER \$20

- SUPPORTS INTEGER, REAL, AND STRING DATA
- 25 STATEMENT TYPES
- GENERATES 8080 ASSEMBLER CODE
- RUN TIME LIBRARY SOURCE AVAILABLE (\$50 EXTRA)
- MANUAL ONLY \$5 (REF. WITH SOFTWARE PURCHASE)
- REQUIRES ASSEMBLER/LINKER BELOW

ASSEMBLER/LINKER \$10

- SYMBOLIC 8080 RELOCATABLE ASSEMBLER
- LINK EDITOR (LIBRARY SEARCH CAPABILITIES)
- LIBRARIAN FOR LINK EDITOR LIBRARIES

FREE BROCHURE AVAILABLE

REQUIRES CP/M-80 V2.2 AND 32K
8" 3740 SSSD OR APPLE 5" 16-SECTOR
DISK FORMATS ONLY

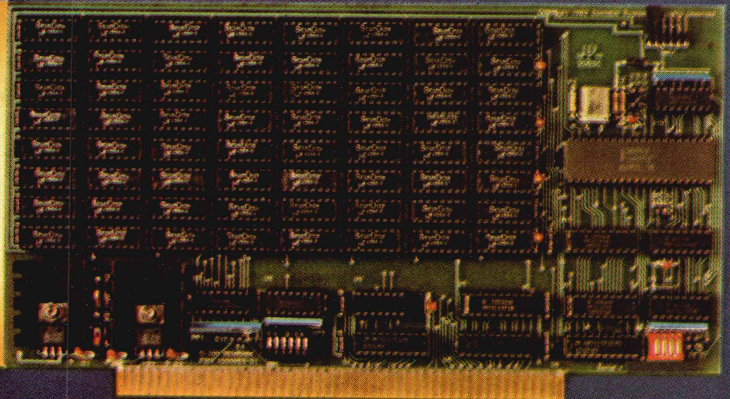
WHEN ORDERING ADD \$10 FOR
SHIPPING, HANDLING, AND MEDIA COSTS.
MASS. SHIPMENTS ADD 5% SALES TAX.
MAKE CHECK OR MONEY ORDER PAYABLE TO

JV SOFTWARE
P.O. BOX 684-NEWTON, MA 02162

CP/M is a trademark of Digital Research, Inc.
APPLE is a trademark of Apple Computer, Inc.

Circle no. 53 on reader service card.

THE PRICE OF **FAST** WAS JUST SHATTERED!



256Kbyte SemiDisk™ **\$995**

For more than a year, we've been making the most advanced disk emulator available for microcomputers. The one that's taken the "waiting" out of computing. Now, we have some more news that'll set the world on fire: A price cut! The NEW 256Kbyte board is only \$995. And the 512Kbyte SemiDisks for the S-100 and TRS-80 Model II are \$1495. (1Mbyte unit is \$2350.) So, what are you waiting for?

The SemiDisk is the **ORIGINAL** single-board microcomputer disk emulator. It has a greater storage density than any other: 1 Mbyte per board! And we've been shipping them for over a year! We didn't do this with 'me too' engineering. Our products are true innovations, based on reliable technology and proven designs, without the need for custom components.

Floppies are ok for data transfer or long-term storage. But they fall far short as online storage. If you are using high level languages, spelling checkers, word processors, databases and other disk-intensive software, you know the price you are paying: time. Your productivity is going down the drain. The SemiDisk disk emulator will save time and increase your productivity.

Even better, Release 5.0 of the SemiDisk CP/M-80 installation software contains SemiSpool, an automatic printer buffer. No extra hardware is required; it's all in the software. Up to 8 Mbytes of buffer space! It's a better solution than a \$350 64Kbyte printer buffer that wastes space on your desk. Send documents of almost any length to the printer at a very high speed, then continue using the computer immediately. No Waiting!

SemiDisk

It's the disk the others are *still* trying to copy.

SemiDisk Systems, Inc.

P.O. Box GG Beaverton, OR 97075 (503) 642-3100

Call 503-646-5510 for CBBS®/NW, a SemiDisk-equipped computer bulletin board.

SemiDisk trademark of SemiDisk Systems, Inc. Copyright © 1983 SemiDisk Systems, Inc.



Circle no. 91 on reader service card.

BRIDGE GRAPHICS

PLOTPAK is a complete plotting library that runs under FORTRAN-80 and performs a variety of functions: windowing, linear print arrays, automatic polygon drawing, annotations, plotting symbol/line selection, labeling, coordinate conversions.

PLOTPAK can drive a screen and plotter simultaneously while maintaining a window which allows you to communicate back to your FORTRAN program. PLOT-PAK includes your choice of the following drivers:

SCREENS

- MicroAngelo MA 512, 520
- ADM + Retrographics
- TEK 4010 Compatible Terminals
- Otrona ATTACHE Portable Computer

PLOTTERS

- H. P. Plotters 7225B & 7470
- Houston Instruments DMP-4

PLOTPAK (.REL file) + two drivers \$190
PLOTAK (source code) + two drivers \$240



Computer Company
DIVISION OF SEA DATA CORPORATION

ONE BRIDGE ST., NEWTON, MA 02158
PHONE (617) 244-8190

DO-IT-YOURSELF CROSS-ASSEMBLERS LANGUAGE TRANSLATORS STRUCTURED ASSEMBLY LANGUAGE ALL IN ONE \$39.95 PROGRAM!

- Use the XLT Macro Processor as a pre-processor to your favorite assembler and do the job right! To translate, a file of definitions is read by XLT and compared to the input text. A match causes the input text to be replaced with the definition's contents.
- XLT allows up to ten arguments in a definition, arguments containing spaces or tabs, incrementing/decrementing strings, a stack, and conditional text replacement.
- Includes definitions to translate Z80 to 8080 mnemonics and vice-versa, and definitions to implement CSAL, a C-structured assembly language, allowing code like `add a,(hl)` to be written `a += *hl`.
- XLT users can pool their definitions via the XLT Users Group.

XLT costs 39.95 plus 5.00 P&H. For CP/M 2.2 8" single density disk Z80 systems.

Send check or money order to:

6708 Melrose
Los Angeles
Calif. 90038
213/932-0817

L.A. Software

CA res. add sales tax CP/M is a trademark of Digital Research

Circle no. 56 on reader service card.

IMPROVED CP/M® TO ATARI® INTERFACE! The Critical Connection™ ONLY \$175

- ★ Automatic install in many systems
- ★ Hardware connects Atari to CP/M system
- ★ Software allows Atari to use disk drives, printer and keyboard of the CP/M system.
- ★ Ability to simulate, with the CP/M system, 4 Atari disk drives simultaneously.
- ★ Software buffers your keyboard & your printer.

The Critical Connection™ received 3 ratings of "excellent" in the Sept. 26, 1983, InfoWorld review.

(408) 997-0264

USS ENTERPRISES

6708 Landerwood, San Jose, CA 95120
Call/Write for Free Brochure

Circle no. 118 on reader service card.

CBASIC* USERS

Now it is possible to recover a "BAS" from an ".INT" file. Send me a SSSD 8" CP/M* disk with a ".INT" file on it: I will return it with the reconstructed "BAS" file added. (Multiple ".INT" files on a disk are allowed, not to exceed 48K per disk.)

Cost is \$40. per ".INT" file. Discount of 10% for 5 to 9, 15% for 10 or more, ".INT" files shipped as a single order.

MC/VISA HONORED • N.J. RESIDENTS ADD 6%

PETER INGERMAN
40 NEEDLEPOINT LANE
WILLINGBORO, N.J. 08046

*CBASIC and CP/M are trademarks of Digital Research, Inc.

Circle no. 49 on reader service card.

80 CHARACTER VIDEO BOARD • WORDSTAR/dBASE II OPTION • TYPE AHEAD KEYBOARD BUFFER



- 25 LINE NON-SCROLL OPTION
- Z80 CPU and 8275 CRT S-100
- CHARACTER GRAPHICS
- ADAPTABLE SOFTWARE
- ORDER ASSEMBLED & TESTED OR PRE-SOLDERED (ADD YOUR IC's)

VDB - A2 bare board from \$49.50

Simpliway PRODUCTS CO.
(312-359-7337)

P.O. BOX 601, Hoffman Estates, IL 60195
add \$3.00 S&H, 3% for Visa or Mastercard
Illinois Res. Add 6% Sales Tax
WORDSTAR is a trademark of MicroPro INTERN'L CORP.
dBASE is a trademark of ASHTON-TATE CORP.

Circle no. 93 on reader service card.

MTS - Z80 Multitasking Operating System and Development Tool

- Handles up to 64 tasks
- Task priorities (1-255)
- Real Time software clock
- Interprocess Communication via Pipes
- Up to 128 FIFO pipes
- Debug monitor/loader
- Process switch overhead under 250 usec. at 4 MHz; proportionally faster at 6, 8 MHz
- Requires 1 interrupt, 4k ROM, 4k RAM, Z-80 CPU
- Z80 assembler source code

MTS-Z80 \$100.00
on 8" CPM Disk

Inquire for Other Formats
Manual \$10.00

Will be applied to purchase of MTS-Z80
Illinois residents add 6% sales tax

CYBERPAK

Box 38

Brookfield, IL 60513
(312) 387-0802



Circle no. 29 on reader service card.

Get the power of your Z80, and the elegance of direct access to CP/M functions from your high level programs with

SYNLIB utility library

SYNLIB consists of MICROSOFT compatible object code that may be called from any high level language that uses MICROSOFT parameter passing conventions.

SYNLIB gives you extremely powerful array and buffer manipulation using the Z80 LDIR instruction; program access to the CP/M CCP console command line; high speed disk block I/O; a high quality random number generator; hex to ASCII conversion optimized by special Z80 instructions; program chaining; and more.

And, because our programmer abhors a vacuum, each 8" floppy comes packed with some of the most valuable public domain software, including available source, absolutely free. You get SWEET, a menued disk utility that makes a computer phobe a systems programmer; UNSPOOL, so you can print and use your computer without buying an expensive buffer; I, to get multiple commands on a line; MODEM7, so that you too can join the free software movement; and many others.

SYNLIB \$50.00 8" SSSD CP/M format
SOURCE: \$100.00

Licensing for commercial use available.

SYNTAX CONSTRUCTS, Inc.
14522 Hiram Clarke, Houston, Texas 77045
(713) 434-2098

CP/M is a registered trademark of Digital Research, Inc.
Microsoft is a registered trademark of Microsoft Corp.

Circle no. 102 on reader service card.

SOURCE CODE

Dynamic Z80 Full Screen Debugger \$30.00
Includes SOURCE CODE. Dynamic full screen update of all registers and specified memory block. you can Single Step, Modify, Set Breakpoints, Disassemble, Display Memory. Use symbols. Too many other features to list here.

Modem Finder - 8080 - Z80 \$30.00
Includes SOURCE CODE. Allows you to specify up to 10 phone prefixes and the starting and ending suffix for each. It will log all numbers with modems on the other end to disk. For Hayes Smartmodems or compatible modem.

Setcom-Zilog SIO Configuration Program \$30.00
Includes SOURCE CODE. Allows setting: Baud Rate, Number bits per word, Stop bits, Parity, Request To Send, Data Terminal Ready, Auto-Enables for automatic hardware handshaking and CP/M's IOBYTE. Baud Rate may be set from 1 - 65535 baud if a CTC chip is used to generate the baud rate.

All orders shipped on 8 in. single density disk unless prior arrangements are made. All prices plus shipping. All orders shipped C.O.D. unless prior arrangements are made. All orders shipped from stock.

Steele & Associates
3328 Fall Creek Dr.
Nashville, TN. 37214
615/889-6636

8080 Trademark of Intel Corp.
Z80, SIO, CTC, Trademarks of Zilog Corp.
CP/M Trademark of Digital Research, Inc.

Circle no. 99 on reader service card.

MEMORY FREEDOM FOR \$99.50 LOVLY-3

The Overlay Linker from easy

LOVLY-3 is a powerful development tool for the implementation of large Fortran programs that exceed the memory capacity of a microcomputer.

LOVLY-3 is a linker with program segmentation and overlay capabilities. It accepts object files generated by MicroSoft's F80 compiler and M80 assembler, and produces executable programs which run under Digital Research's CP/M operating systems.

- No modification of the source program being segmented; implicit loading of segments.
- No limit on total program size; segment size up to 64k.
- Multiple entry segments: up to 200 modules; 8 level deep overlay structure.
- Complete user documentation with example.

LOVLY-3 runs on Zilog Z80 and Intel 8080/ 8085 based micros with 56K memory and CP/M 2.x.

Disk formats include: 5" NorthStar Advantage and Apple II. Other disks available soon.

Terms: Check or COD.

easy

Engineering Application Systems
P.O. Box 10998 #529
Austin, TX 78766-1998

Circle no. 36 on reader service card.

EXPECT A LOT *from* VEDIT

The Professional's Editor for Program Development, Word Processing, Source Code Translations

Widely acclaimed as an editor, VEDIT has evolved to be much more. Only VEDIT offers the combination of a versatile full screen editor integrated with a powerful command language based on TECO. For the first time you'll be able to perform complex, yet useful, text manipulations that are virtually impossible with other editors or word processors.

VEDIT is completely customizable - it easily configures to your favorite keyboard layout. Use any function or cursor key you wish. Plus, our total commitment to hardware support ensures that VEDIT will be perfectly matched to your individual needs and to any microcomputer you are ever likely to own.

VEDIT cuts programming time in half - with multiple file handling, macro capability and special features for Pascal, PL/1, 'C', Cobol, Assembler and other languages. And it can help with source code translations (example ZILOG to/from INTEL translator macros are included). A complete line of translators will be available by the year's end.

Word processing is a snap with word wrap, paragraph and print functions. Command macros free you from tedious search/replace operations. Hundreds of search/replace on dozens of files can be performed by VEDIT without waiting or intervening.

Go ahead and expect a lot from VEDIT. Its performance and our support will satisfy your most exacting needs.

VEDIT is available for virtually every 8080, Z80 or 8086 microcomputer made. To order please specify your computer, operating system and disk format.

VEDIT - Disk and Manual

8080, Z80 or IBM PC	\$150
CP/M-86 or MSDOS	\$195
Manual only	\$18

VEDIT's FEATURES:

True Full Screen Editing	Text Move and Copy	Word Wrap, Format Paragraph
Horizontal Scrolling	10 Scratchpad Buffers	Simple Printing
Edit Files One Disk in Length	Load/Save Buffers on Disk	150 Page Indexed Manual
Automatic Disk Buffering	Powerful Command Macros	Startup Command File
Compact (Only 16K) and Fast	Directory Display	Menu Driven Installation
Display of Line and Column	Edit Additional (small)	Program CRT Function Keys
Set/Goto Text Markers	Files Simultaneously	Support Newest CRT terminals
'Undo' Key to Restore Line	Insert Another Disk File	Support Smart CRT functions
Automatic Indent/Undent	Unlimited File Handling	Flexible Memory Mapped Support
Adjustable Tab Positions	Recovery from Full Disk	Customizable Keyboard Layout
Repeat Function Key	Change Disks While Editing	

CP/M is a registered trademark of Digital Research Inc. MSDOS is a trademark of Microsoft. IBM is a trademark of International Business Machines.

© Copyright 1983 by Popular Computing, Inc. a subsidiary of CW Communications, Inc., Framingham, MA—Reprinted from InfoWorld.

InfoWorld

Software Report Card

VEDIT 1.36

	Poor	Fair	Good	Excellent
Performance	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Documentation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Ease of Use	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Error Handling	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Quotes from InfoWorld review by Tim Daneliuk of last year's version of VEDIT

'The performance of this product is nothing short of outstanding!'

'The ability to customize the editor is especially useful.'

'VEDIT is a 'virtual' text editor. The amount of memory in your computer does not limit the size of the file you can edit.'

'VEDIT acts just as its advertising and documentation claim, with no unexpected errors or other problems. If

anything, the advertising claims for this product are understated.'

'This is a well-conceived product. The many Visual-mode control and escape sequences make editing text simple, and the Command-mode features greatly simplify file handling and memory management.'

CompuView

PRODUCTS, INC.

1955 Pauline Blvd., Suite 200 Ann Arbor, Michigan 48103 (313) 996-1299

ADVERTISER INDEX

Reader Service No.	Advertiser	Page No.	Reader Service No.	Advertiser	Page No.	Reader Service No.	Advertiser	Page No.
1	ABC Data Products	107	40	Forth Technology	31	79	Plum Hall, Inc.	37
2	Adler Computer Systems Co.	68	41	Gladstone Electronics	73	80	The Programmer's Shop	105
3	American Scientific Comp. Corp.	15	42	GGM Systems	75	81	PRO Microsystems	107
4	Altecon Research Group	105	43	GTEK, Inc.	111	82	PRO Microsystems	86
5	Ariel Corporation	53	44	Hallock Systems Consultants	101	83	Puterparts	56
6	Ashton-Tate	121	45	Harvard Softworks	59	84	Quelo	105
7	Autocode	2	46	Hawkeye Grafix	111	85	Quest Research	41
8	Awareco Active Software	62	47	Hawkeye Grafix	95	86	Quic-n-Easi Products	114
9	BD Software	73	48	Hudson Marketing	49	*	Edward Ream	77
10	Blat, R & D Corp.	117	49	Peter Ingerman	124	88	Redding Group Inc.	99
11	Bonnie Blue Software	43	50	Integral Quality	58	89	Revasco	107
*	Borland International	7	51	International Software Alliance	96	90	Sage Computer Technology	127
*	Bridge Computer Co.	124	52	Introl Corp.	115	91	SemiDisk Systems	123
14	California Digital Engineering	8	53	JV Software	122	92	Shaw Labs	64
15	Carousel Micro Tools, Inc.	89	54	Key Micro Systems	68	93	Simpliway	124
16	C/Craft	103	55	Laboratory Microsystems	3	94	SLR Systems	61
17	Chromod Associates	91	56	L.A. Software	124	95	Software 99	85
18	The Code Works	96	57	LEDS Publishing Co.	61	96	Software Publishers, Inc.	93
19	Components Express, Inc.	69	58	Leo Electronics, Inc.	36	97	Solution Technology, Inc.	119
20	CompuTalker	83	59	Logical Systems	99	98	Solution Technology, Inc.	117
21	Computer Design Labs	25	60	Loki Engineering	27	99	Steele & Associates	124
22	Computer Friends	103	61	Manx Software	6	100	Symbolic Systems	85
23	Computer Innovations, Inc.	47	62	Mark Williams & Company	11	102	Syntax Constructs, Inc.	124
24	Computer Resources of Waimea	86	63	Master Computing, Inc.	16	104	Techné Software Corp.	118
25	Compuvision Products	125	64	Mega-Byte	87	106	Telecon Systems	35
26	C-Systems	29	65	Micah	99	108	Tesseract Associates	77
27	C User's Group	79	66	MicroMotion	71	110	Thoughtware, Inc.	66
28	C Ware	57	67	Micro Processors Unlimited	75	112	Trantor Systems, Ltd.	101
29	Cyberpak	124	68	Microtype	56	114	Unified Software Systems	77
30	Data Access	97	69	Micro Ware	108	116	Unir Corporation	113
31	Dedicated Micro Systems, Inc.	55	70	Modular Mining Systems, Inc.	89	118	USS Enterprises	124
32	Dedicated Micro Systems, Inc.	109	71	Mycroft Labs, Inc.	95	120	Visual Age	75
33	D & W Digital	90	72	Nimble	51	122	Western Wares	53
34	Disco-Tech	85	73	Overbeek Enterprises	55	124	Whitesmith, Ltd.	65
35	Dysan Corp.	4	74	Overbeek Enterprises	109	126	John Wiley & Sons	13
36	Easy	124	75	J.D. Owens Associates, Inc.	36	128	Workman & Associates	6
37	Ecosoft, Inc.	79	76	Pascal & Associates	23	130	XyQuest Inc.	45
38	Epson America, Inc.	128	*	Peterborough Book Services	63	132	Yourdon Press	17
39	Floppy Disk Services, Inc.	39	78	Phact Associates	51			

Announcing the Winners!

Questionnaires were mailed out to a random sampling of *Dr. Dobb's Journal* subscribers. Those who completed the questionnaire were eligible for a raffle. A list of the prize winners follows. Thanks to all of you for completing the questionnaire. This annual reader survey is a valuable tool in planning the coming year.

One DDJ Bound Volume Set

Jack R. Davis, Knoxville, TN

DDJ Bound Volume 6

*M. L. Aitel, Herndon, VA
Arthur L. Greenberg, Yardville,
NJ
John Lobenstein, Santa Clara, CA
Edward A. Malin, Peekskill, NY
A. F. Niessner, Jr., Boalsburg, PA
Wm. W. Richardson, Sherman, TX*

Dr. Dobb's T-Shirt

*Steve Brooks, Minot, ND
John W. Campbell, Bozeman, MT
Roger L. Cole, Elmhurst, IL
Norman Dresner, Severn, MD
Daniel J. Feiling, Lake Jackson,
TX
Steven J. Flap, Winona, MN
Richard Gerth, Abilene, TX
Ron Herbst, Terre Haute, IN
Michael R. Jankovich, Salt Lake
City, UT
William M. Kline, Miami, FL
A. Kontos, Houston, TX
Jeffrey K. Lines, Collinsville, IL*

*Roland V. Lupient, Mosinee, WI
G. McNevin, Richardson, TX
Michael McCarter, Philadelphia,
PA
Steve Monroe, Herndon, VA
Jerry Owen, Hopewell, VA
Robert Pierazek, Riverside, CA
Daniel Scalici, Detroit, MI
Glen A. Spooner, Oceanside, CA
Larry D. Thompson, Ann Arbor,
MI
Richard E. Thurmond, Whittier,
CA
Robert J. Wallace, Raleigh, NC
Gordon White, Maquoketa, IA
David Whip, Reistertown, MD*

Congratulations!

When it comes to software development, the difference between a Sage IV computer and other micros is like day and night.

With the Sage Computer it'll take you fewer days and nights to finish your program.

Speed is the reason. Speed resulting from the fast MC68000 microprocessor, fast architecture, fast operating systems, and blinding transfer rates.

The fact is, even we are surprised by the amount of software that's developed on Sage Computers.

So if you're a programmer, maybe you should spend some time learning how much time you could save using a Sage IV.

The Hottest Languages And Operating Systems.

Programs in nearly all of the important languages can be written on the Sage computer.

Included in the price is the **p-SYSTEM** operating system which supports **Pascal, BASIC, and FORTRAN.**

Other operating systems are optional. For **UNIX** fans there's **IDRIS**, which runs up to twice as fast (even without a Sage computer). **IDRIS** conforms to /USR/Group Standards Committee

standards and programs written under it are highly portable to other micros.

CP/M advocates please note that Digital Research has developed **CP/M-68K** for Sage hardware, providing a truly complete software development environment. Versions of **Pascal, BASIC, C-BASIC, C, and FORTRAN 77**, as well as a very fast **APL**, may be used under this operating system.

The Sage IV is also blindingly fast when

If you found your language here, you just found your computer. The 16-bit, 2-million operation/sec Sage IV micro with up to one MByte RAM and 18-MByte internal hard disk.

For more information and the name of your nearest SAGE dealer, call us today. And be sure to ask about our new software catalog describing over 200 application programs for Sage computers.

Sage Computer Corporate Office,
4905 Energy Way, Reno, Nevada
89502. Phone (702) 322-6868
TWX: 910-395-6073/SAGE RNO

Eastern United States
Sage Computer
15 New England
Executive Park
Suite 120, Burlington, MA
01803 (617) 229-6868

running **hyperFORTH** with its extended programmer and user interfaces.

Besides **IDRIS**, other **Multi-User** operating systems that run on the Sage Computer are **PDOS, MBOS, and MIRAGE.**

A log of excitement has been brewing in the Pascal World over Niklaus Wirth's new **MODULA 2**—and it's available for Sage computers. So is **ADA**, for the record.

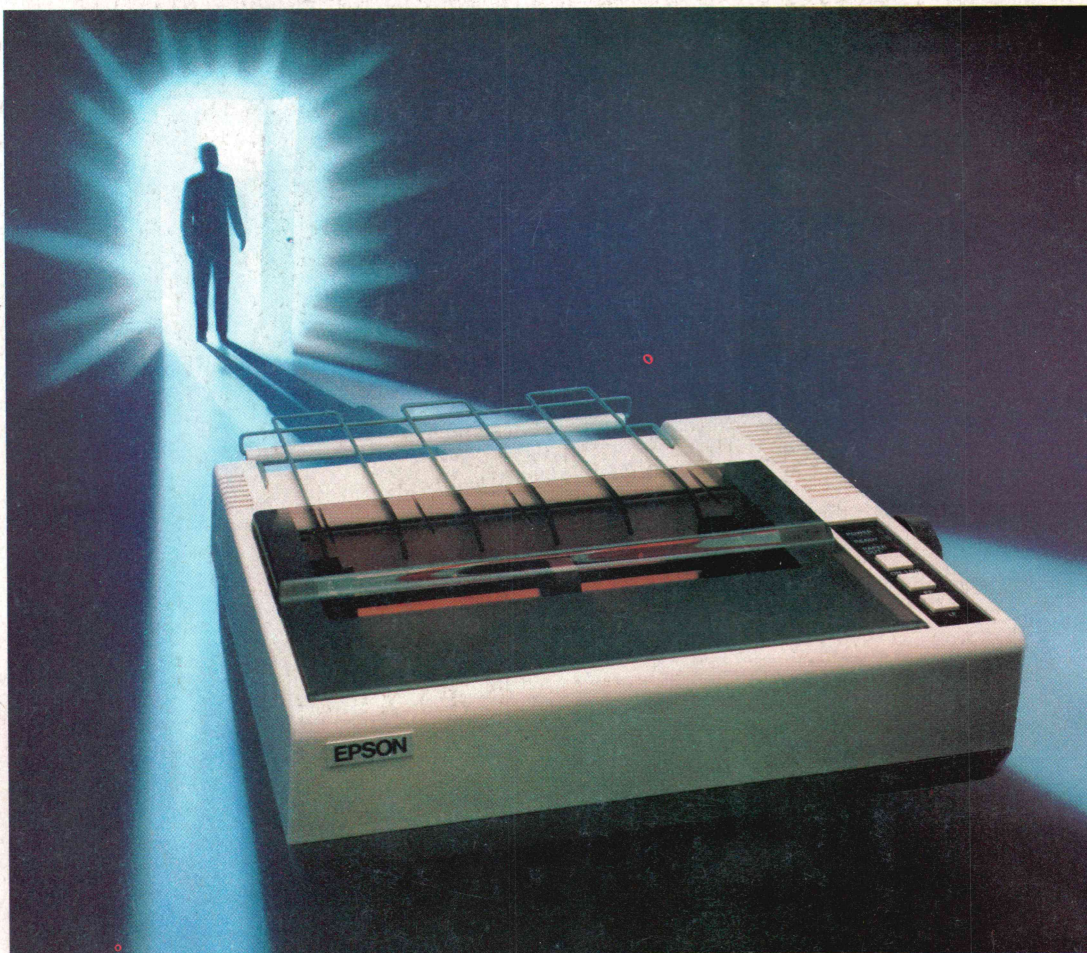
© 1983 Sage Computer all rights reserved
Sage & Sage IV are trademarks of
Sage Computer

SAGE™
COMPUTER

The computer you don't wait for!



Introducing the new RX-80.



A printer this good
could only come from one company.

Epson.

Take a look at the new Epson RX-80 with a critical eye and you'll know what we mean. Epson printers are put together with manufacturing tolerances in the micron range. And meticulously checked for quality — one at a time.

Our reliability rate is approaching perfection. Which means you'll probably never — ever — have a problem with your RX-80.

Frills at a no-frills price.

Just because the RX-80 is affordable doesn't mean we've left off the goodies. You get 128 different typesstyles with a resolution so sharp you have to look twice to make sure it's dot matrix. You get bidirectional printing, logic seeking, and a speed of 100 char-

acters per second. And you even get programmable forms control, graphics, and a special Quiet Mode to keep the noise down.

This is your printer.

No matter what computer you own — or will own — do yourself a favor and make sure your printer is an Epson RX-80.

We build them to be the finest printers on earth. No matter how small the price.

Number one. **EPSON**
And built like it. EPSON AMERICA, INC.

3415 Kashiwa Street, Torrance, CA 90505

Please call (800) 421-5426 for the dealer nearest you. In California (213) 539-9140

Circle no. 38 on reader service card.